asileFX

snowFlaKe

(the MetaNode Collection v2.0)

Tools and Enhancements for Vue 7+

Users Manual

This manual last updated on:
June 9 2009
(See user manual release notes on page 54)

http://www.asilefx.com

Installation

snowFlaKe *(the MetaNode Collection v2.0)*

There are several ways to access the tools available with *s*now*FlaK*e, each with their own advantages and disadvantages. All installation methods have been provided in separate folders, depending on the method you prefer.

For the quickest access to the tools in this collection, the Host File Access method is suggested. If tools are commonly used; you can use the other installation methods to add nodes to the menus and quickly access them.

Host File Access – *snowFlaKe.mnd or snowFlaKe_upgrade.mnd*

With *s*now*FlaK*e, a new all access MetaNode is available. This host MetaNode is a structured directory, with additional MetaNodes contained within it. You can use the "edit" command, or double click, to open the MetaNode and see its contents. After choosing the tool you would like to use, select it, and use the copy and paste commands to load it into the desired function/graph.

The host file may be loaded into any function/graph for quick access to the tools and setups. Using the "Menu Access" method is suggested for the Host File. You can select any menu you like.

Menu Access - *Content_Install*

The Vue Content directory, contains a directory named MetaNodes; with 11 directories within this directory. Each primary node type in vue has its own directory. When right clicking in Vue's Function Editor, these primary nodes are shown in a pull-down menu. Within each menu you will find a MetaNode sub-menu. Any MetaNode(s) saved in the content directory MetaNode folders, will be accessible through the sub-menu. If no metanode files are saved in the content directories, then the sub-menu will not be shown in the primary node menu.

The *Content_Install* folder is setup for direct access to the MetaNodes within the MetaNodes sub-menu when in the Function Editor. This involves mixing the asileFX collection with your custom and pre-installed MetaNodes.
To avoid any deletion of current nodes, a manual drag and drop, or copy and paste, of the individual files into these folders is suggested.

Single Menu Access - *Content_Install_OneGroup*

Another option for installation, is to chose one of the primary node folders to host the entire asileFX MetaNode Collection. This will allow you to access all nodes from one sub-menu; for this method an empty folder is recommended. The default installation of vue has several primary node groups without MetaNodes, and any of them can be chosen. For example, no "Combiner" MetaNodes are installed by default. To access all asileFX MetaNodes though the MetaNodes sub-menu for Combiner Nodes.
Copy all Nodes from the *Content_Install_OneGroup* folder, into the content install directory \MetaNodes\Combiner\.
With the Function Editor Open, Right Click in the node window. Under the Combiner menu, you will find a sub-menu called MetaNodes, will all the asileFX MetaNodes contained within it. Technically a MetaNode is a combiner node, so this is appropriate.

Upgrade Installation

Upgrading from the MetaNode Collection v1.0 to snowFlaKe ***(the MetaNode Collection v2.0)***

The upgrade to *snowFlak*e includes only the new tools available in v2.0. You may add the new tools to your existing installation method. Node folder structures were updated in version v1.5, and match those of the v2.0 release.

It is strongly suggested that you download the v1.5 update before combining collections.
The update is available from your asileFX.com account, on the order history page.

If you received the MetaNode collection as part of the Ultimate Vue Training and Content Bundle, or from a reseller, please contact asileFX to setup the download.

You can also contact asileFX to download the full version, you will be sent a download link once your previous purchase is confirmed.

The asileFX Website: http://www.asilefx.com or http://www.asilefx.net

Users Manual Contents

Page #

Installation ..... [**2**

Getting Started ..... [**5**

Using the Host File: snowFlaKe.mnd 5

Universal Tool Controls 6

Master Controller Operations 7

Node Index ..... [**51**

Release Notes ..... [**54**

snowFlaKe ***(the MetaNode Collection v2.0)***

Featured Tools ..... [**8**

Setups and Helpers ..... [**23**

Helpful Information 23

Node Setups 24

Color Operations ..... [**26**

Color Generators 26

Color Correction 27

Dynamics ..... [**36**

Constraints and Controllers 29

NGOS - NGOC 35

Numeric Operations ..... [**20**

Control Enhancement 36

Filters and Combiners 36

Terrain Filters 39

Math 40

Texture Map Operations ..... [**41**

Texture Map Setups 41

Conversions, Dynamics, and Controllers 43

Vector Operations ..... [**45**

Composer Modifiers 45

Conversion Operations 46

Decomposer Modifiers 47

Offset Operations 48

re(de)composers 48

Vector Math 49

Vector Splitters 49

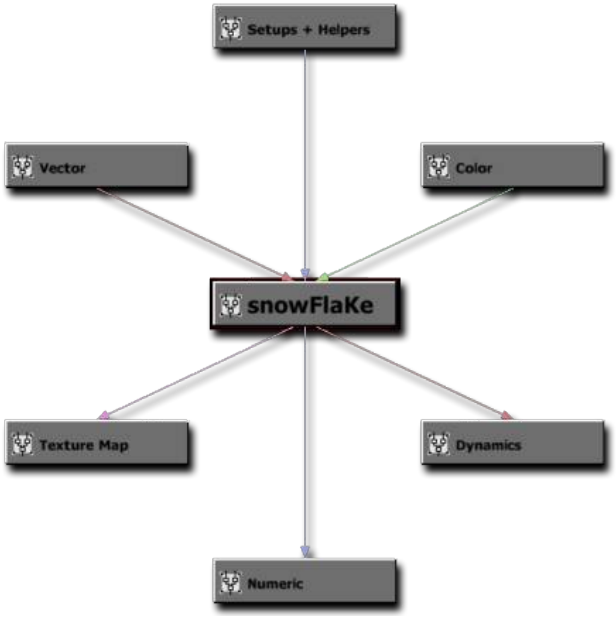
GETTING STARTED

Using the host file:

snowFlaKe.mnd

The best way to load any tool or setup into your functions or graphs, is through the host file. All nodes and additional setups are available in this master host node.

You can double click on the “asileFX snowFlaKe” node to open it, or click on the edit button.



Each of these nodes found in the host file are directories for the tools in the collection.

Tools and setups are sorted into the manual's categories, some with slightly different names.

snowFlaKe	Featured Tools
Setup + Helpers	Setups and Helpers
Color	Color Operations
Dynamics	Dynamics
Numeric	Numeric Operations
Texture Map	Texture Map Operations
Vector	Vector Operations

Copy and Paste Multiple Nodes

To make a multiple selections of nodes, or nodes and their setups; you can left click and hold down the mouse button to create a selection box over the nodes.

When more than one node is selected, the copy button at the top of the interface is grayed out. You will need to use the keyboard shortcut for the copy command. The Paste command button will normally work, if not, use the keyboard shortcut.

Keyboard shortcuts can be found under File>Options in the *Operations* tab.

Custom Node Layout

By using the above methods you can customize the host file. Just make sure you save the host node after making changes, although I would not suggest overwriting the original file.

Universal Tool Controls

Some settings and controls can be found in many tools. Rather than explaining these "universal controls" in every node description; the following is a list of these commands that exist in several nodes, and what they are for.

Switches On / Off Sliders
0=(*x*) 1=(*y*)
These switches are found mainly in newer nodes, or updated nodes. They are intended to be used without an in- between value.

Many inputs or conversions need to be enabled to work. Be sure to look for these switches and see what they control. In most cases the name after the = will say what it does. All switches are found in Groups, separations between controls, the group name will almost always list what the switch controls.

How a switch works internally:
The more complex the function, the more calculation time needed to complete the output.

Unit Conversion – Length to Vector
In almost every node this conversion affects only manual input amounts (typing in a number). These entries control a position or size vector.

Vector Unit
A pure vector. The distance this unit will extend will vary depending on the internal unit. Direct connections between vectors are always matched vector units. A direct connection would be controlling the position of one object with the position of another object. Since no manipulation or manual entry is needed for these connections, the vector unit is almost never seen, and since Vue displays length units rather than vectors, the amounts appear consistent.

Length Unit
Length units are the unit of measurement, defined by the selection of options in the Units and Coordinates Tab, File>Options. These are the units displayed in the interface. While the tool descriptions most often refer to a length unit of meters; any display unit may be chosen, km, feet, etc.

Manual entries of a length unit in the main interface, are automatically converted to the correct vector unit. However these units in functions will vary. Defining position, size, or origin is always a vector. Vue has two primary definable vectors, Vector 3 and Vector 2.

Vector 3 (Vector 3D) is an x, y, z vector; this is known as just Vector.
Common Input/Outputs:
Position, Orientation, Size, Pivot Offset, Offset, Normal.

Link Color: Red
3 Numbers as a three channel input/output

Vector 2 (Vector 2D) is an x, y vector; this is a Texture Coordinate.
Common Inputs/Outputs:
Texture Origin, Image (Texture) Offset, Image (Texture) Scale, UV Position.

Link Color: Purple
2 Numbers as a two channel input/output

Master Controller Operations

Several tools in this collection have master switch or control options. These allow multiple connections between graphs. Tools with these operations will have a special dependent node, used to retrieve the data being sent. All of these dependencies will require connections to an External Dependency input node. These setups are provided in the host file, with the External Dependency input nodes named to the dependency required.

All external dependencies will start with [ ] before the name. With the exception to the color controller, which starts with [c].

Some master controls, such as vector or texture coordinates, may not use a unique dependent node to retrieve the information. In most cases these dependent nodes are used to send number outputs. To learn more about why these controllers are required to retrieve number outputs, see the description for NGOS and NGOC.

Master Controllers MUST be used in graphs or a terrain procedural, or the external dependency [] will not show.

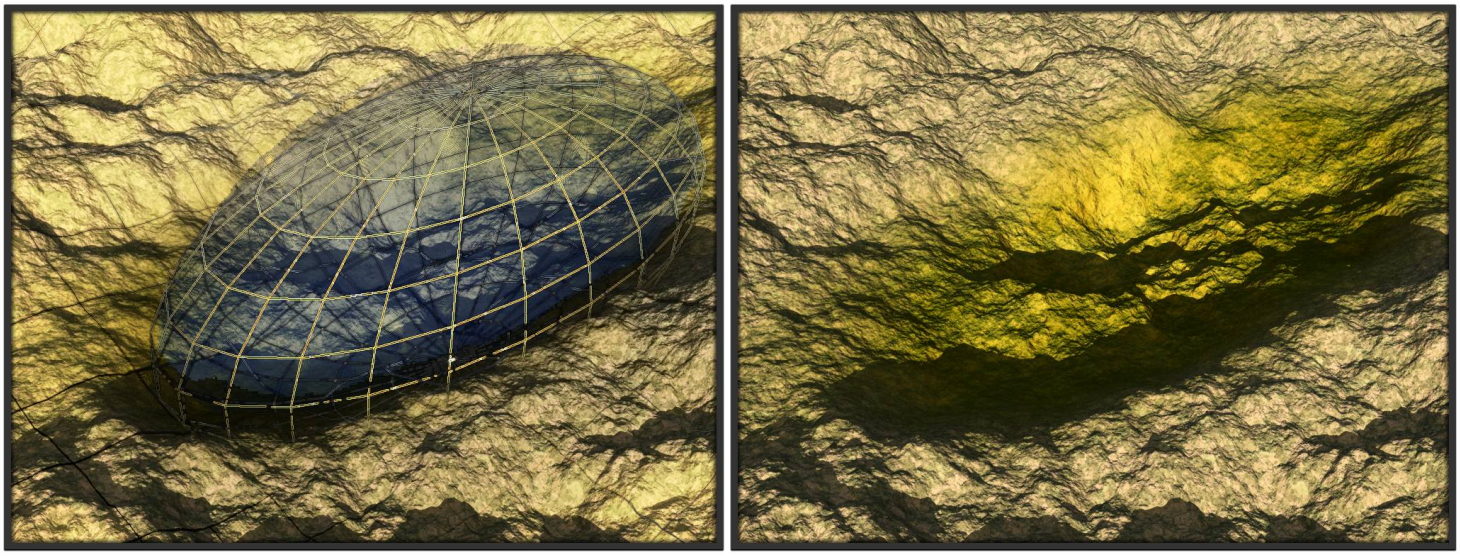
FEATURED TOOLS

Advanced Sphere Zone

Advanced Sphere Zone is an easy use dynamic number/density controller. This node is extremely powerful and has a rather complex internal function. The original Sphere Zone, is part of the asileFX Vue 7 Training – Advanced Production Techniques – Optimization and Dynamic Population. Since the training, the controller has been completely redesigned and rebuilt. The new improvements add a whole new level of flexibility and control to your scenes.

So just what is Advanced Sphere Zone?
Advanced Sphere Zone is a dynamic tool used to control a volumetric Number / Density / Altitude function output.
The "Zone" exists in material space, but is controlled by a sphere object. The scale and position of the sphere object are matched to the material. All x,y,z scales are independent, so a perfect match is possible.
Additional adjustments can also be made within the node, to scale smaller or larger than the controller object. The threshold and falloff of the material can also be adjusted.

The only limitation is that the controller sphere's orientation(rotation) can not control the material. Velocity of material origin is also not available within the node, but can be matched.



Left: **Render / Wireframe / openGL - Mixed** *Right:* **Render**

In the above rendered scene, Advanced Sphere Zone is used to control the subtraction of one fractal to another with a procedural terrain.
A mixed material's distribution is directly linked to the zone, controlling which material shows when the zone is intersecting with the terrain. You could expand on this scene by linking the density of an ecosystem layer to the zone. All these options can be animated by changing the size or position of the sphere. By using Advanced Sphere Zone with the "Dynamic Range Controller," you can also animate any control within the node options.

Creating a Sphere Zone

[In the Scene]
The Controller Object (Sphere)
Add a sphere to the scene. Rename controller, or whatever you like.
You'll want to setup a special material for the controller, so that it does not show up in renders, or affect ecosystem population. A material is provided with the collection, called controller if you would like to use it.

Note: Any object can be used as a controller, but because the material space is a spherical, the sphere object is appropriate. However, Infinite planes, billboards, terrains, and planes are not at all suggested.

[In the Function]
Open the function you wish to control, and load the "Advanced Sphere Zone" MetaNode.

Connecting the Controller

Integrated External Dependency Nodes are known to cause problems in dynamic functions, because of this the dependent inputs must be created manually; unless you use the setup from the snowFlaKe.mnd host file.

Add in two "External Dependency" *Input* nodes.
From the Dependency pull down menu in the input node. Select the Position output of the controller object in one node, and name it Position, and in the other node select Size, and name the node Size.

The top of the Advanced Sphere Zone node has two vector 3(red) inputs. They are named size and position.
Connect these to the External Dependency nodes you just created, size to size, position to position.

The controller object is now connected and controls the material. *See "External Settings" below to continue the setup of the controller.*

Node Settings

Controller Z Position
0=Null 1=On [Slider – on/off switch]
This slider turns on and off the controller Z (up axis) position control. By default the Z position is used and is recommended. Turning this control to Null, allows you to use the Vector Input at the top of the node, called "Z Position if Null." You can use this to define where the material is positioned on the z axis. This allows you to hover the controller above or below the desired material space. You can also use another object to control the position of the z axis. Only the vector from the z axis is used, x and y are completely null, and controlled by the controller position.

You can also use this slider to blend the distance between the z vector input (0 if nothing is connected) and the Z position of the controller.

External Settings *(Node settings)*

The settings in this group are for matching the material vectors to the controller vectors. These controls are extremely important, and with exception to the "Material Origin," these settings exist outside the function editor.

Material Scale This is the scale of the material layer, shown in the upper left corner of the material editor. Or under the cloud or procedural terrain material preview if editing a cloud layer or procedural terrain.
If controlling a mixed materials distribution, the scale will be the top layer of the mix.

Function Scale *(Output Scale)* All primary outputs in the *advanced material editor* have a "Function Scale." They are used to scale the connected function outside of the function editor. This scale does not change the settings in the function itself, but instead scales the output of the function.

The following outputs have a Function Scale in the material editor. **If connected to one of these outputs, the Function Scale must match.** Several scales have different names, listed below are the scale names matching the output.

Node Output ***(Function editor)***	**Function Scale** ***(Material Editor)***
Color	Function Scale
Bump	Bump Scale
Highlight	Scale of highlight function
Transparency	Function Scale
Reflection	Function Scale

Distribution	Scale of Function
Scaling	Function Scale
Color	Function Scale
Density	Function Scale
Output	Scale

Material Origin This value is listed in the Effects Tab of the material editor, under "Origin of Material."

Fine Tuning

Size This slider increases and decreases the scale of the material. This is a relative value, and not a length unit. It is useful for small tweaks to the scale.

Threshold The material is a total descending gradient, the threshold increases the inner density of the output. This feature is directly connected to the Falloff and gain, adjusting their total influence.

Falloff Adjusts the gradient falloff, or blending edge of the volume. Falloff of 10=Small Edge, 0=Very Smooth Blend.

Gain Increases the total number output, multiplied by the falloff amount.

For a very sharp edge, use the Vue Threshold Filter or Smooth Threshold Filter, connected between the output of sphere zone and the controlled. If the output of the zone is beyond the current control range, normally -1 to 1 as a number output, you can use the clamp filter node, with low at -1 and high at 1.

Invert Output

0=Normal 1=Inverted *[Slider]*
This slider will reverse the output of the node, it can also be used as a gain output.

Subtraction Roughness

0=Off 1=Input On *[Slider on/off switch]*
The top of the node contains a "Subtraction Roughness" input node (number). This can be connected to any number output, like a fractal or noise node, to add variations within the volume.
This feature should only be turned on if the input is connected. You can also use this to turn off the roughness without removing the connection to the input.

Note: The Invert Output option will also invert the subtraction.

Material Coordinates

Internal Uses the options under "Position type." Default is Internal, World – Standard.

Input (vector) A vector input called Material Coordinates, allows you to input a functions Position or Normal inputs, or use Position Options Input. The advantage of this is, you can modify the material vector before it reaches the node.

Zone Dependent

The Zone Dependent node allows you to make multiple connections to the same sphere zone. The sphere zone node must be contained in an object graph or procedural terrain function, or the dependent will not be accessible. By using and External Dependency Input node, you can select "[] Sphere Zone" from the list for the control object. Connecting to the Zone Dependent gives you the ability to define some of the output settings, specific to the current function. If using the controls in the dependent node, the "external tuning" number output will display the local results. To use the settings from the main node, connect to the Sphere Zone number output.

Camera Master Controller

Vue 7.5+ Recommended

This is an extremely useful controller for camera projection and camera locking.

The node is intended to be loaded into a camera's graph. Once loaded into the graph you can enter a camera's position, orientation, focal length, and focus distance, into the node; then connect the camera's outputs into the node to lock them in place.

The node will also host the projection output to materials, as an external dependency of the camera.

Unit Conversion – Vector to Length

Affects Position and Focus Distance inputs.

External Dependencies (Master Controller)

This node features master control outputs. See (Master Controllers) for more information about using these options.

Master Outputs:

[ ] Camera Projection [ ] Scale [ ] Position [ ] Orientation [ ] Controller Dependent

Camera Controller Dependent

This node is used to retrieve number outputs from Camera Master Controller.

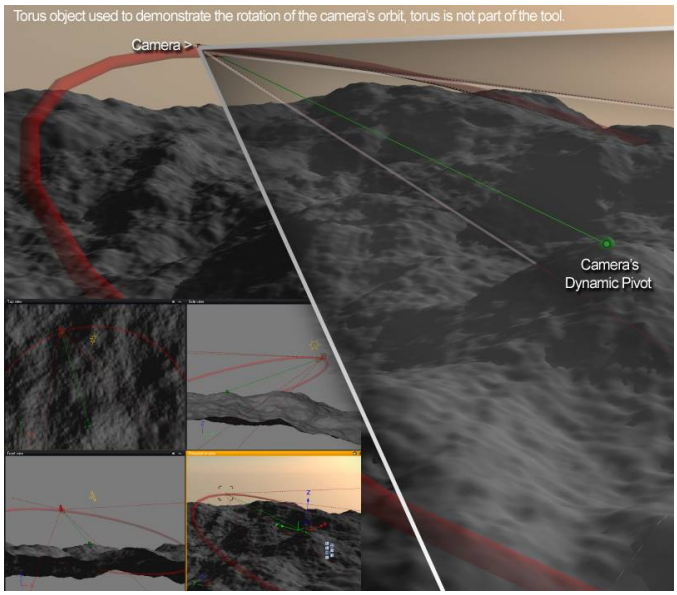
Connect the input of this node to an External Dependency input node.
Dependency: [ ] Controller Dependent

Outputs

Aspect Ratio	Focal Length	Focus Distance

Camera Orbiter

Camera Orbiter will project a camera's pivot point in front of the camera, to the center of the view; dynamically adjusting itself as the camera moves. This allows you to rotate the camera around elements in the scene. Currently Vue's in-viewport camera control, is limited to a tripod style rotation; you can use the rotate around selected objects command, found in the navigator, but this is limited by the selections pivot location. If the pivot is off screen, as it is when close to a terrain, this become very hard to control.


Torus object used to demonstrate the rotation of the camera's orbit, torus is not part of the tool.
Camera
Camera's
Dynamic Pivot

Note: Definable position or size units are Vector units. To minimize the calculations required to operate this tool, and reduce accidental errors; no unit corrections are integrated into Camera Orbiter. What this means is unless you are using a scene internal unit of 1, all entered values will need to be multiplied (or divided in some cases) by the internal unit, unless stated otherwise. In most cases you can extract the value and use the conversion node for external calculations outside of Camera Orbiter.
[Read more about the internal unit, and vector vs. length]

Extremely Important:

When using Camera Orbiter, NEVER use the Move or Pan camera controls in the render preview. This normally breaks the camera. If you do break the camera, the undo command will normally correct the action. The rotation and focal length controls do normally work.

Its a good idea to save the main camera to create a duplicate, then apply orbiter to the copy. If the camera breaks, you can fall back to the original main camera.

Turn off Lock height above ground – this feature is currently not compatible.

Note to 3Dconnexion hardware users:

When using a 3Dconnexion mouse or similar navigation device with Camera Orbiter. You must turn off the "Pan Right/Left" controls. This option can be found in the Advanced Settings tab of the 3Dconnexion control panel. Using the pan controls will break the Vue camera. Camera Orbiter was designed for Orbit / Spin movement around objects. If you need to pan, use the mouse, Shift+Right Mouse Button.

Orbiter will also cause the "Pan Up/Down" controls to reverse. You can reverse this option in the control panel to correct the movement.
It is also suggested that you turn up the Z Movement Sensitivity in the Camera Orbiter options when using these devices.

Inputs / Outputs

Camera Orbiter has two inputs and two outputs. All inputs and outputs are Vector 3, meaning they using x, y, z position and orientation data.

All connections can be made in the camera's graph without the need to add additional external dependency input nodes. *Note: The default input nodes of all object graphs are external dependencies; data must first travel from the outputs, and then back to the inputs.*
All inputs need to be connected to Camera Orbiter before orbiters outputs are connected.

Inputs

Position Connects to the camera's "Position" *input* node.
(Vector) Required for: All modes and usage.

Orientation Connects to the camera's "Orientation" input node.
(Vector) Required for: Pitch Zoom Control.

Outputs

Position Connects to the camera's "Position" *output* node.
(Vector) Required for: Position Constraint

Pivot Offset Connects to the camera's "Pivot Offset" *output* node.
(Vector) Required for: All modes and usage.

Camera Orbiter has two modes, fixed and dynamic.

Fixed Mode

This mode uses a fixed amount to control the distance of the pivot projection.

Example Distances :
(Note: If using "Pitch Zoom" the fixed distance will change with camera pitch.)

if Length Unit = Meters
(Approximate Distance, very close though)

Internal Unit (.1)
Fixed Length (100) = 10 meters

Internal Unit (1)
Fixed Length (100) = 100 meters

Conversion:
Length Unit = (Fixed Length x Internal Unit)

Dynamic Mode

The mode uses vector inputs to control the distance of the projected pivot from the camera.
Making a direct connection to the camera's position is suggested. If you run into any problems with this connection, use the Sun Light's position as an alternative, although depending on the location of the sun light, results can vary. The Sun light is always connected to the camera, following along and staying at a fixed distance; making it almost equally useful for the connection.

Dynamic Adjustments *(These settings have no affect on the fixed mode)*

Z Movement Sensitivity: Increase or decrease the sensitivity of the camera movement, the larger the scene, the higher the sensitivity can be without extreme camera movements.
(Default .2)

Length: This unit is the length the pivot can move past, or below ground level, (z vector 0). It will also stop the dynamic calculation when the camera reaches this height. The default unit of 4 normally does not need to be changed.
Not an exact unit.

Ground Height: Terrains, or the main object of focus, are often positioned above the ground plane, or Vector Position Z (0). Ground height allows you no compensate for this offset, by subtracting this amount from the Length Projection. This often prevents the camera from intersecting with objects.

if Length Unit = Meters

Internal Unit (.1)
Ground Height (100) = 10 meters

Internal Unit (1)
Ground Height (100) = 100 meters

Conversion:
Length Unit = (Ground Height x Internal Unit)

Terrain Average *(Can be used with the Dynamic and Fixed mode.)*

These settings allow you to define a terrain's attributes to calculate the ground height. The calculations and number of outputs needed to compensate for the differences in Z scale for every terrain type, would interfere with the performance of the Camera Orbiter. Therefor, the following two settings need to be entered into the node, and no external dependency is used. Not only does this method increase performance, its much quicker.

Click on the numerics tab with a terrain selected to find the following information.

0=Ground 1=Terrain Average Choose to use the "Ground Height" from the *Dynamic Adjustments* tab or the terrain average.

Z Position Z Position of the terrain

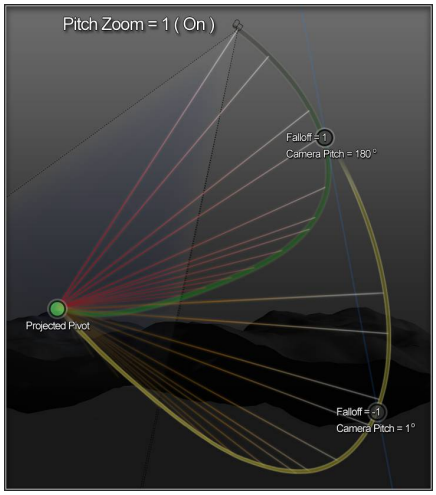
Z Size Z Size of the terrain

Divide Size by Default (3) this value works quite well for most terrains, but can be changed if needed.

Unit Conversion:
Length Unit = (Z Size x Internal Unit)

Pitch Zoom *(Can be used with the Dynamic and Fixed mode.)*

This feature will move the camera closer to the pivot as the pitch changes, basically acting as a zoom; it will not affect the camera's focal, and it is an extremely useful way to navigate.



0=Off 1=On Turns the Pitch Zoom feature on and off, and also can be used to decrease the distance of the zoom.

Reverse Changes the pitch direction for the zoom.
At 0 the camera will move closer to the pivot as the camera pitches toward the sky.
At 1 the camera will move closer to the pivot as the camera pitches down to the ground.
At .5 this feature is turned off.

Zoom Falloff This slider adjusts the amount of pitched required for the zoom to occur.

Z Position Constraint *(Stops camera from moving below ground or defined ground amount.)*

Camera Orbiter prevents Vue's built in feature, "Stop camera going below clipping plane (ground / water)" from working; the Position Constraint feature is used to fix this, with a few added bonuses.

0=Off 1=On Turns this feature on and off.

Ground Level Sets the point where movement on the Z axis will stop.

0=Ground 1=Terrain Average Either use the entered ground level, or use the Terrain Average calculation, or a mix of the two.

Note: This feature will not prevent the camera from orbiting below the constraint.

Dual Control

It is possible you use both the fixed an dynamic modes at the same time. To do this, use the on/off slider. While at 0 the dynamic mode is fully active, and at 1 the orbiter uses the fixed amount. Moving the slider to a value between 0 and 1 will use both modes, proportionate to the distance to the controls at 0 or 1.

Keep in mind that the fixed value is mixed with the dynamic value, meaning you can adjust the fixed value for even more control of the dynamic output.

Camera Animation with Orbiter

It is possible to animate the camera with orbiter, and doing so will create some very interesting camera movements. One thing to keep in mind; the displayed animation path will be the pivot if using the global coordinate mode, and the camera itself if using local coordinates. Although sometimes the path seems kinked, like it may jolt the camera, the animations are normally smooth, provided there aren't too many drastic changes between keys. Playback and renders of animation paths seem consistent. However, when the animation is paused in the interface, during the timeline, the camera may bounce to a different position. This normally only happens when using pitch zoom, although the correct dynamic calculation occurs when it needs too.
Has not been render farm tested. I strongly suggest using Orbiter to plan out the animation keyframes, recording the vectors, and manually keying a different camera.

Tips and Suggestions:

By displaying the camera's pivot can watch the dynamic changes, or see the fixed length. The point of focus is also useful to judge the length of the pivot from the camera. The camera will need to be selected to see the pivot.

Keeping the camera selected while navigating though the scene can reduce performance, keep this in mind if using the selection to see the pivot.

Connecting the position output of object to an external dependency of the camera's position, locks the object to the camera's projected pivot, changing dynamically and providing a good reference.

Position changes to the Orbiter Camera outside the main camera view; not suggested with Pitch Zoom on.
Rotation changes to the Orbiter Camera outside the main camera view; use local coordinates.

Navigation: The best way to navigate with Camera Orbiter in the camera view.

Zoom

Using the mouse scroll is the best way to move in and out of the scene. Free rotating scroll wheels (ones that will continue spinning until stopped) work wonderfully for this. You will need to make sure that “Zoom on mouse wheel” is turn on in the Operations tab File>Options.
Although called “Zoom” its really moving the camera and not adjusting the focal. Adjust the focal Ctrl+Right Mouse Button.

Using the Pitch Zoom and Dynamic mode, can be the best ways to control camera “Zoom.”

Translate Camera (Panning) + Zoom

Another way to move in and out of the scene “*zoom*” with Camera Orbiter, is to use Translate Camera movement, and move the mouse up and down. To translate the camera, hold down Shift+Right Mouse Button.

This is also how you can move the camera back and forth.

Dynamic Range Controller

In-depth creation and usage of this node (v1.0) is covered in asileFX Vue 7 Training "Advanced Production Techniques – Dynamic Function Control and Animation."

This node is used to control number outputs and origin operations by using an object in the scene. When controlling numbers, you can chose which axis transformation of the control object affects the numeric change. By connecting a number to the output of this node the axis can be chosen.

The node has a built in vector correction for the internal unit. (See the "Internal Unit Vector Correction" node for more information about the internal unit.) Even when using this node to control origin the internal unit must be set to correctly calculate the range output.

What's so great about this node?
This node allows you to control and animate any extractable output in any function, with precise control. Just about anything can be animated in the function editor with Vue's default tool set, by using the "Time" input node, however controlling this number input can be rather difficult and does have its limits.

Settings

All inputs are Length Units, not Vector units, regardless of the internal unit. With the exception to the Distance Control: Start and End Values, which are the custom inputs. Setting the internal unit to 1 will turn off the conversion.

Controller Starting Position (Offset = Distance Control: "Start Value")

Offset [Start location (position or size vector) of the control dependency object]

Controller Axis Range (Distance = Distance Control: "End Value")

XYZ Distance Distance control dependency object must move on the chosen axis output, to reach "*Distance Control: End Value.*"

Vector Distance Distance control dependency object must move on all axises, when vector is chosen as the output, to reach "*Distance Control: End Value.*"

Distance Control

Start Value Output amount when controller object is at its default location

End Value Output amount when controller object travels to the desired XYZ or Vector Range

Unit Conversion Length Input to Vector Output *(for Vector)*

Scene Internal Unit The scene's internal unit, used to control all inputs and outputs as Length Units and not Vector Units.

Inputs

Controller (Vector) Connect to an external dependency input node to act as the controller object. Use a Position or Size vector from the control object to control the node.

Outputs

X Distance Control, controlled by changes made to the dependency's X axis.
Output controlled by "Controller Axis Range: X Distance"

Y Distance Control, controlled by changes made to the dependency's Y axis.
Output controlled by "Controller Axis Range: Y Distance"

Z Distance Control, controlled by changes made to the dependency's Z axis.
Output controlled by "Controller Axis Range: Z Distance"

Vector Distance Control, controlled by changes made to all dependency's axises.
Output controlled by "Controller Axis Range: Vector Distance"

Setup Example

First you'll need to create an object, or pick an object in the scene to be the "controller." This is the object you will be using to control the output, by changing either the position or size of the object.

Within the function you want to control, load in the "Dynamic Range Controller," and add an "External Dependency" input node. In the pull down menu of the External Dependency node, select your controller object and chose it's position or scale.

The vector input on the top of the Range Controller Node (the small red arrow pointing up) can now be connected to the External Dependency.

As an example, let's say we have two Fractal nodes, connected with a Combiner node, on the blender setting. The combiner has a extractable ratio control slider, which is a number input. The connections from the combiner node to the fractals are "Input 0" and "Input 1." If the ratio is set to 0, then Input 0 is fully shown, if the ratio is 1, then the fractal connected to Input 1 is fully shown. The in-between values 0 – 1, is the percentage of the blend, at a ratio of .5, 50% of each fractal output is shown.
The controller object can be used to control the ratio of the combiner node. If the position output is chosen in the external dependency, you can move the controller between point a and point b to change the ratio.

In the Range Controller, under Distance control, there are two number inputs, Start and End Value. By default the start value is 0 and the end value is 1. This also happens to be the values needed to control the ratio, the high and low value. If you were controlling the "Roughness" of a fractal, let say from .25 to .6, then the "Start Value" would be .25 and "End Value" would be .6, or reversed depending on the direction the controller is moved.

Extract the ratio from the combiner node. Once the ratio is extracted, a number input is connected to a constant number node. Replace the connection from the constant node, and move it onto the bottom of the Range Controller node. A drop down menu will appear with the available outputs, x, y, or z; the gray output is for controlling a vector, and in this case, is not needed. Select the output for the axis you want as the control, for this example, let's say X.

Set the internal unit of the scene, under "Unit Correction," by default this is (.1). To find what your current scene internal unit is, go to File > Options. Under the Units and Coordinates tab, the internal unit setting can be found.

The "Controller axis range" is the distance the controller object will need to move to go from the start value to the end value. Since X is connected, under X type the range, in units. So, for 20 meters, type 20. Now as the object moves 20 meters on the X axis from its start location, it changes the ratio, from 0 – 1.
So at 0 meters the number output is 0 and at 20 meters the number output is 1.
The Vector Distance can be used when using the controller for vectors, the range is a total range for all axises.
The Controller Starting Position, allow you to set the start value of the controller. This amount is a unit and not a vector value. With an internal unit of .1, a vector offset value for 25 meters would be 250, but a unit value is 25.

Additional Techniques:

It is also possible to extract the start and end values and connect them directly to nodes, then connect the control node directly to the output. In relation to the above example, you could remove the combiner and connect the two fractals directly to the Start and End Values.

Range Controller - Dependent

This dependent control can be loaded into the controller object's graph. It will constrain the position or size of the controller object, to the distance entered in the Dynamic Range Controller. The constraint is for negative and positive range, so the distance the object can travel is doubled. Uses Master Controller Operation rules.

Inputs

Position Connects to the position or size of the controller object's position or size input node.

[] Range Dependent Connects to an external dependency input node. Dependency: [] Range Dependent

[] Range Dependent 2 Connects to an external dependency input node. Dependency: [] Range Dependent 2

Outputs

Position Connects to the position or size of the controller object's position or size output node.

Function Preview Scalar

This tool changes the scale of node previews shown in the function graph. Since the Output Observer only shows the final output of a function, you often need to look at an individual channel to see what changes are being made. If you are working with a World Coordinate system, on a function with very large features, a fractal may no show you the details you need to see.

The node essentially changes the scale of all nodes in the function equally, by modifying the Vector Input nodes.

Vector Input
Connect to the Position or Normal Input nodes of a material.

[ ] Master Switch (Vector Input)
Find out more about the master switch below.

Vector Output
Connect all nodes connected to the Position or Normal Input nodes to the Preview Scalar Output.

Full Vector Offset
This is a built in Vector Offset for the material, it is used at all times, even if the preview is off.

Preview

Offset	Changes the offset position of the preview.
Off = 0 On = 1	Turns the previewer on and off.
0=Local 1=Master Switch	Turns the previewer on and off using a connection to the Master Switch input, instead of the local off = on slider.

Zoom

0 = In 1 = Out	Choose whether to zoom in or out of the preview.
Zoom Amount	Increases the zoom (scale) of the preview.
Fine Tune Amount	Adjusts the sensitivity and range of the Zoom Amount slider.

The Master Switch

This node uses the master switch node to control, Preview Scalar on/off

If editing multiple materials with the previewer, you can create a master object to turn all previews on and off with the master switch. The Master Switch node can only be loaded into an object graph or procedural terrain.

How to use the Master Switch and create a master Dependency object:

Add an object into the scene (the sphere works nicely.) Open the sphere's graph and load in the “Function Preview Scalar – Master Switch.” Connect any of the available outputs to the Function Preview Scalar node (I suggest orientation, in case you want to move or scale the master object. The output is null and isn't used to control the sphere, so, it really doesn't matter which output is used.)
Also consider using the “hide from render” and “ignore when populating ecosystems” options for master control object (sphere).

In the material functions using the preview scalar, add in an External Dependency input node; from the pull down menu, select the sphere, and the [] Master Switch dependency.

Now connect the vector input on top of the node, called [] Master Switch, to the External Dependency.
In the preview scalar, under the preview group, change the “0=Local 1=Master Switch,” slider to 1.

At any time you can now open the Graph for the master object, and turn on or turn off all the preview scales connected to the switch.

Matte Painter - 2 Layer

Vue 7.5+ Recommended

This node features a complex camera projected texture map setup. It allows you to setup a foreground and background image, with two alphas. One alpha controls the foreground image, and one controls the subtraction of all layers. In addition you can fade the foreground image, to blend with the background. This node uses a full alpha subtraction outside the projected area, this clips the image outside the camera projection.

Texture Repeat Methods

Texture repeat methods are set to the following:

Mirror X and Y:	None:
Foreground, Alpha, Background	*All Layers Alpha*

Camera Master Controller

Allows the node to be connected to the Camera Master Controller node to control the projection. To use, add the external dependency input node, and select the camera with the master controller loaded, select [] Camera Projection as the dependent.

The following settings will be controlled by the master controller and will no longer work in the node.
Aspect Ratio – Width and Height
Image Tweaker – Scale
Camera Projection – Projector

Image Aspect Ratio or Pixel Resolution

(See the node "Aspect Ratio" for more information)
All layers are controlled by this ratio.

Image Tweaker

Allows you to scale and position all layers outside the projection area. If the aspect ratio is defined, these controls are not needed for an exact match of the projected camera area.

Image Interpolation

Allows you to set the interpolation methods for the Foreground, Alpha, and Background layers.

Camera Projection

Projector – Choose the camera to project all layers.

Image Layers (Mattes)

Background This is the bottom layer of the setup. If you are only using one image layer, use this one.

Foreground This is the top layer of the setup. The opacity of this layer can be adjusted with the Foreground Fader – Amount slider.

Alpha This map subtracts the foreground to show the background.
Important:
By default without an image loaded, the preview will be white, or inverted. This allows you to use the Foreground and Background with the Foreground Fader, and without the alpha. After removing a loaded image, with the trash can icon, the preview becomes black, and the foreground will no longer show above the background. Use the invert image icon, upper right of the image map, to invert and correct the layers.

All Layers Alpha

This image will subtract from all image layers as part of the "+All Layer Alpha" output. The interpolation method of this image is located under the image, and not part of the interpolation group.

Foreground Fader

Amount – This slider will adjust the opacity of the Foreground image.

Node Outputs

Color (color)	This is the primary output. It contains the Foreground and Background, plus any Alpha (Foreground) blends.
Clip Outside Projection (number)	This output can be connected to the Alpha output node of the material function. It will cut out all parts of the image not contained within the camera projection.
Clip Outside Projection +All Layer Alpha (number)	Adds the all layer subtraction alpha to the outside projection clip.

Important Notes and Techniques

Alpha and Foreground Fader
By default without an image loaded, the preview will be white, or inverted. This allows you to use the Foreground and Background with the Foreground Fader, and without the alpha. The Foreground Fader - Amount slider will not work if no image is loaded into the alpha and the preview is black. After removing a loaded image, with the trash can icon, the preview becomes black, and the foreground will no longer show above the background. Use the invert image icon, upper right of the image map, to invert and correct the layers and fader.

Turning off the Foreground
Above it is mentioned that the Foreground will not work without the alpha inverted, or an image loaded in.
By inverting the alpha without an image loaded in, it is possible to turn off the foreground layers. If the Foreground Fader amount is set to 0, it is also possible to swap the layers, by inverting the alpha without an image loaded in.

Under normal usage the Foreground can be turned off simply by moving the Foreground Fader - Amount slider to 1.

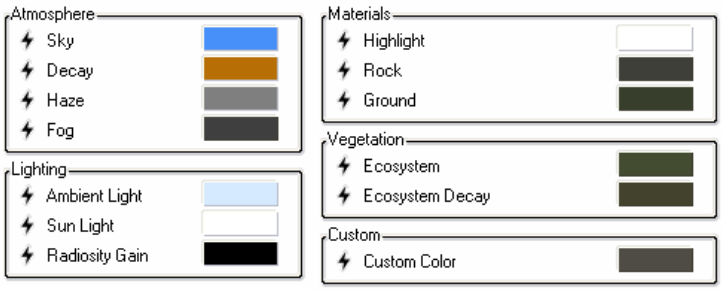
Outside Projection Clip (Alpha)
The All Layer Alpha is designed to work with rendered alpha output, meaning black is removed and white stays. Because of this it is important not to invert the All Layer Alpha image within Vue. When inverting an image it also inverts the unseen or outer bounds of the image; inverting the image within vue would cause the alpha output to work incorrectly. If the alpha needs to be inverted, use an image manipulation program to invert and save the changes.

Scene Color Controller

Although a fairly simply tool, it can prove extremely useful for global color changes. What this node does, is host several groups of customizable colors for the scene, from an object graph. Load this node into any object function, preferably a null object; a null object can be any object that does affect the scene. You can use an basic object, hide it from the render, make ecosystems ignore it, and hide from the G-Buffer. A few objects have been provided for this purpose, and can be found in the included objects folder.

After loading the node into the null object graph, connect one of the vector outputs to the node. This link is here as a precaution, to make sure Vue knows it is there. If a node is just floating, unconnected to any outputs, the graph may not be recognized. This vector has no other purpose than this.

When selecting the node you will see the following options.



Each color can be changed or extracted.

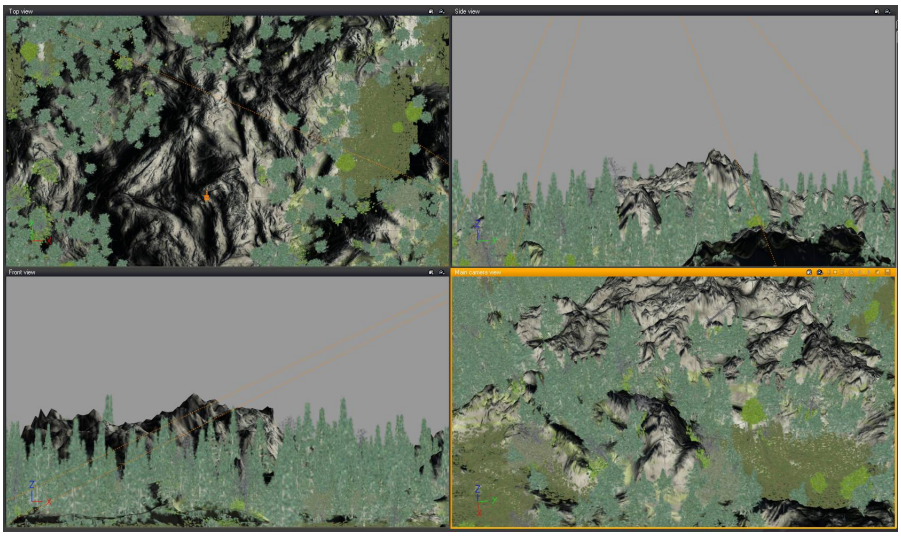
Now the important part, and point of this tool. All colors defined in this node can be accessed in any function or graph. By adding in an external dependency input node, into any function, you can select the null object as the dependent, and a list of every color is available.

All of these colors will have [c] in front of the name, to separate it from other outputs.

You can now setup global colors for the scene. If at anytime you want to change the color, open the host graph, modify the color; anything and everything connected to that color will change. Even if it's a thousand materials.

Why is this a featured node?
It's not the node itself but the concept in its creation. This node was chosen because it is so simple, easy to use, and modify. Not to say that this node isn't powerful because of its simplicity, but it is unlocked and additions can be made, making it a great dynamic learning device, and a tool.

Terrain Previewer - Camera Controller



This tool will allow you to render previews of terrains and objects. The connected dependency must be proportionate on the x and y scale vector to work correctly. To use this node. Save a copy of the Top Camera, name the new camera Terrain Previewer, and open the object graph. Load in the Terrain Previewer MetaNode.

This node has two vector outputs and one number output. Connect the two vector outputs to the camera graph's corresponding outputs, Position and Orientation. Now connect the Focal output to the nodes focal output.

Next we need to chose the object we are creating the preview for. Set the Size and Position to the size and position of the preview object. Standard and Procedural terrains exist in different coordinate systems, the Height / Procedural slider corrects this difference. All other objects in the scene are equal to the space of a procedural terrain, with the exception to the plane object type. For a standard terrain, set the Height / Procedural slider to 0, for a procedural terrain, this slider needs to be set to 1.

Because the focal values needed to create an almost flat, non-perspective preview, do not render well from that distance. The material preview will not be perfect. Because of this, 4 options are included in the node to fine tune the camera extents. These sliders are extremely useful when previewing terrains that are very high, having a scale on the z (up and down axis), higher than 10% of the x, y scale. Or terrains with large features, or all features above the height of the pivot position.

If you take a moment to scale your preview object, you will notice the position of the preview camera will remain locked to the outer bounds of the preview object.
The position of the camera should not to be manually changed, instead use the 4 sliders in the camera controller node. X and Y shift will move the camera on the x and y axis, based on a percentage value of the terrain size.
The “Move Sensitivity” sets the percentage relative to the object size, and controls the sensitivity of the X and Y shift sliders; increasing or reducing the amount of movement from the 0 – 1 value.
“Focal tweak” will move the camera on the z axis, with a relationship that also changes the length of the cameras focal. It is recommended that you have either the front, side, or both viewports visible when adjusting these sliders.

To render the preview, set your output render ratio to 1:1 square, in the render options, and define what you would like rendered. Depending on the size of the terrain, you may need to remove all fog settings from the Atmosphere Editor, or set the aerial perspective as low as it will go.

You can now make a copy of the terrain (to preserve the original material) then apply the render to the image map option of the copied terrain material. Using the object parametric mode will eliminate the need to scale the material.
Rename this material so it is not the same as the source terrain.

Procedural Terrains are known to display openGL images reversed on the axis. Basically displaying the image as if you were looking at it from below the terrain. Corrections will need to be done in an image editing program to fix this. You can use the Flip Horizontally option in any image application, to fix the image for previewing purposes. If using a standard / height terrain, the image will not need this reverse flip, but may need to be rotated 180 degrees.

To optimize the scene, set the preview terrain to be hidden from the render, then set the source terrain to remain in the render but be hidden from the openGL views.
If you do not have access to Photoshop and the like, a free open source program is available, and in some ways is better than most commercial applications.
It's called GIMP and can be found at http://www.gimp.org

SETUPS AND HELPERS

Helpful Information

These MetaNodes can be used for quick setups of common tasks and speeding up work flow. All of the following nodes are meant to be "ungrouped", as they have no inputs or outputs.

Ungroup MetaNode

You can extract the contents of these MetaNodes by using the "Ungroup MetaNode" command, located in the top tool bar of the function/graph editor.

Note: Extracting the contents of a MetaNode can sometimes reposition your current function layout.

To avoid this repositioning, you can first edit the MetaNode, moving the nodes off to one side, away from the inputs and outputs. The input and output space is normally the same. If you current function is directly between the inputs and outputs, you can edit the node and move them all to the right or left. Keeping in mind that all nodes are placed under the inputs and above the outputs. Shifting the inputs and outputs within the metanode may shift the inputs or outputs when "ungrouped."

Another method, and normally a better method of extraction, is using the copy and paste commands. See the *Getting Started* section of this manual for more information.

When creating MetaNodes the addition of a Double Output node is almost unavoidable. This node will not be extracted into your function/graph, and is not connected to anything in the node. This output is a number output, named "Null," and can be ignored.

Depending on the function a MetaNode is edited/opened in, a selection of input nodes will show at the top of the function. These inputs will not be extracted when using the "Ungroup MetaNode" command.

You can expect to find the following *null* inputs for each type of function, when opening/editing the MetaNode.

Materials and Density Functions:

Position	Normal	Altitude	Slope	Orientation	Time

Object Graphs:
Extra input nodes normally do not display in object graphs.
If they do, they are the common inputs, lights and cameras will have many more.

Position	Orientation	Size

Procedural Terrain Functions:

Position	Time

Setups

The file names of all setup nodes begin with “Setup_.”

Bunch o Dependents

A collection of named External Dependency Input nodes. When adding external dependencies input nodes, the chosen dependents are not shown in the name; when adding lots of dependents this can be time consuming and confusing if nothing is named. This node adds 12 external dependency input nodes to your function/graph. Names may be changed and unused nodes deleted.

External Dependency *Input* node names:

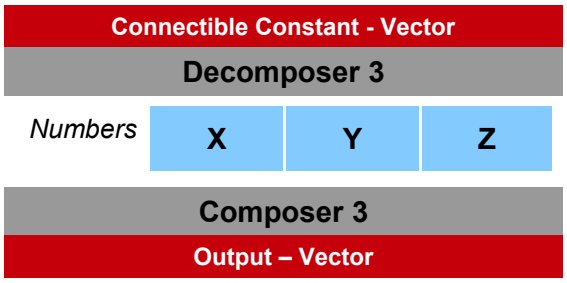
Position	Orientation	Size	Pivot Offset	Offset	Z
Controller	Origin	Roughness	Color	Custom	Custom 2

re(de)composed Vector-Texture

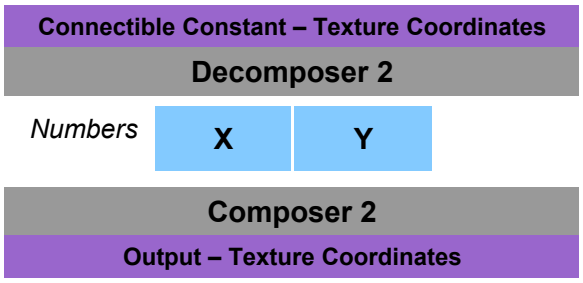
This setup is useful for extracting the number outputs of vectors and texture coordinates.

Includes the following two node setups.

Vector Setup



Texture Coordinates Setup



Custom Dependents

6 Custom Dependency Output nodes.

Custom 1	Custom 2	Custom 3	Custom 4	Custom 5	Custom 6

Combiner Blender

A very useful setup for the Blender | Combiner node. Connectible constants are used to separate inputs, and allows for an easy addition of filters. Filters have already been added to between the connections. These filters will not affect the output, since they are the default filter unchanged. The naming system used, makes it easier to connect other node inputs to the setup, before or after filters.

Blender - Number *MetaNode Setup*

Blender - Color *MetaNode Setup*

Node Groups

This MetaNode contains additional MetaNodes, you can edit the main node, copy ctrl+c any of the contained MetaNode groups, and then paste them into your function for "ungrouping."

The following MetaNodes are contained in this primary MetaNode.

External Dependents 6 External Dependency Input nodes. Named:

Position	Orientation	Size	Pivot Offset	Offset	Controller

Number – Filter Chain 6 Connected filters. All are the default filter, but can be changed to any available filter for numbers.

Vector - Math Chain 6 Connected math nodes. All nodes are Offset nodes by default, but can be changed to any math vector operation.

Math - Chain Undefined chain of 6 connected math nodes. By default all are the Multiply math node.

Constants All 8 constant nodes in one group. 4 Constants, 4 connectible constants.

re(de)composed Same setup as the "re(de)composed Vector-Texture" node, without the connectible constants.

Color Correction All 9 color correction nodes, connected.

Custom Out 6 Custom Output nodes. Named [] Custom 1 – 6.

Blender - Color The *Number* "Combiner Blender" setup node.

Blender - Number The *Color* "Combiner Blender" setup node.

COLOR OPERATIONS

Color Generators

2 Map Color-Brightness Variation

This color generator includes the controls of the Color-Brightness Variation node, with the color defined by two color maps. It also has a mapping mode control, allowing you to change the position options of the internal fractal.
As of version 2.0, this node now features a number input for more control of the color map range.

Color Map Reversal

This node adds a couple of features to modify the number input of the Color Map node.
Located directly above the color map is the Reversal feature.

Color Map

Map Defines the Color Map, controlled by the number input.
Default Color Map is set to Mirror clamping modes.

Reversal Inverts the number input, essentially flipping the color map. Values between 0 and 1 will create an offset of the color density, increasing the distance between colors.

Input Adjustment

Reduce Height Divides the number input to reduce number inputs with a high gain. The default setting is 1, which has no affect on the input, values higher than 1 and -1 will reduce the output or gain. Negative values will invert the number, just like the reversal. Values between -1 and 1 will increase the output, no longer making this a Reduce Height affect, but an Increase Height affect.

Offset Allows you to move through the number input, and fine tune. A negative value will lower the number, and a higher value will raise the number.

Turbulent Color Variation

This color generator has an embedded turbulence fractal, allowing more control over color variation. Many of the hidden controls in the native Color Variation node, have been linked to this turbulence. Using this node requires a link to either a Position or Normal input node. This node has two outputs; The color output is the primary output, if you wish to use this variation for a number link, say the bump channel, it also contains a secondary Grayscale output.

Leaf Color Randomizer

This node is a mix between a color generator and corrector. It adds extra color variations to the color output, based on the existing color.

To use, connect this node between the color output node and leaf texture node or color node.

Off / On Switch

Can be used to compare differences between the original leaf color and the randomized color. You can also use the slider to reduce the influence of the color variation.

The Variation Mapping

Type Used to set the mapping mode of the variance. The default mode is set to Object – Standard, and is suggested for modifying plant species. Global modes are not suggested when animating plants, primarily because the leaves would move through the variance, rather than the variance moving with the leaves.

Includes an additional Grayscale output, converted from the color output before the Darken amount. This output can be used to modify additional outputs, like Bump and Highlight.

Color Correction

Color Darken

This node darkens color by modifying hues, saturation, contrast, brightness, and blending colors, using a rather complex internal function to assist in the modification of natural color elements.

Controls (Sliders)

Darken Sets the overall darken amount.

Desaturation Removes additional color, but is not a direct saturation amount.

The creation of this node spawned two primary variations, one sometimes working better than the other depending on the color variance. Rather than having two separate nodes that do basically the same thing, they were combined into a single node. The Darken slider controls the darken amount of both variations, and the Desaturation amount is actually the blend between the two separate nodes. An amount of 0 outputs the first node, and at 1 it outputs the variation. It's called Desaturation because the variation has less color saturation.

Easy Color Changer

The easy color changer is fairly straight forward, it's an easy and quick way to change the color of a color channel. By using different color values you can define many aspects of the color change, without destroying the original tones or variations of the original color. The more luminous the color, the more transparent the color overlay is. The more saturated the color, the more intense the color change is. You can also use the color as either black or white to darken or lighten the color. The defined color is also extractable, and not limited to a solid color.
The contrast slider can be changed to reduce the color variation and range of the correction color; for the best preservation of tones, a value of 1 is normally best.

This node also contains a Color to Brightness conversion output, named grayscale. This allows you to attach any number node to this output to control variance based on the color changes and output.

HLS Shift Intensity Ranged

Vue's default HLS Shift color correction node can be very sensitive because the slider range is so high. This will often result in the over sampling of colors and too much tweak time.
This is a much more predictable version of the HLS Shift color correction node. The added intensity slider will adjust how sensitive the Hue, Luminosity, and Saturation sliders are to change. Also, the range output from this node is far less than the original HLS Shift.

This node also contains a Color to Brightness conversion output, named grayscale. This allows you to attach any number node to this output to control variance based on the color changes and output.

HLS Shift Super Intensity Ranged

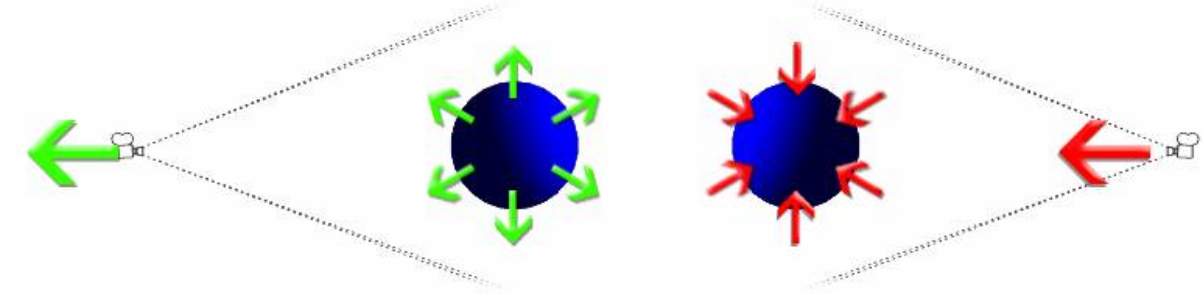
Same as the HLS Shift Intensity Ranged node, will an additional “Super Sample” slider. Super sample will allow the oversampling of the HLS Shift, producing a much higher ranged output. With super sample, this node is closer to the default HLS Shift, but with more control.

DYNAMICS

Constraints and Controllers

Distance = Size

This node will increase a vector amount(size) based on the position of a any object as a dependent.



The node was designed primarily to be used with object size in relation to a camera's distance to the object. With this setup, an object will remain the same size in the camera view, regardless of the camera's distance to it. As the camera moves away from the object, the object size increases. What is seen in the camera view, is the entire scene shrinking, while the object being controlled stays fixed.

The camera distance = size method is extremely useful for controller, reference, master, and null objects. The position of the controlled object is open with this method, which makes even more useful for dynamic relationships. It could also prove useful with Camera Projected Texture Maps.
This node can be used for many other setups as well. If you connect the distance dependency to another object in the scene to control size, the result would be:
Object a = Size Controlled
Object b = Dependency

With Object a in a fixed position, moving object b closer to a, decreases the scale of object a.
As object b moves away from object a, object a gets larger.1

Relative Size Controls

Size *(number)* This is the relative size amount and not an actual unit length or vector unit. This slider is null, if the relative size is switched to off.

0 = Off **1 = On** *(slider)* This slider is used to switch on the relative size, sending the true vector distance to the output. When set to on, the relative scale is used, and can be changed with the Size slider.

Inputs

The two Vector Input nodes, are used to calculate the distance.

Position *(vector 3)* The node's "Position" input connects to the controlled object's position, basically a self position. You can use the default Position Input node of the objects graph to connect this.

Dependent *(vector 3)* The "Dependent" vector input, connects to an external dependency input. In the external dependency, selecting the dependent object's position, is required for positional distance. Although any vector with technically work.

Output *(vector 3)*

The Vector output is the calculated distance between the two vector inputs, to the relative or true vector amount.
You can use this to control any vector output, but the node was designed to be connected to the Size Output node.

Improved Distance Constraint

This node is used to limit the distance of a vector. It works just like the default dynamic distant constraint, but this version has a couple of new features added to it.

Distant Constraint

Center The vector origin.

Min Distance How close the object can get to the center.

Max Distance How far away the object can move away from its origin.

Free Axis 0=Constrained 1=Open

What makes this node "Improved," is the ability to turn of the constraint on any axis.
This group contains three sliders (x,y,z) to turn on and off each axis constraint.

When a slider is set to 0 then the constraint is applied to the axis. If the slider is set to 1, then that axis is ignored from the constraint, and the objects position is added to the center.

Master Switch

This node is a universal master switch for off/on sliders. This node can be used both as a sender and receiver.

Output

off/on This output is connected to any off/on slider. It is only used when the node is the receiver. And the dependent input is connected [] Master Switch.

External Output

[] Master Switch Can be found in an External Dependency Input node.

Master Switch - 3

Just like the normal master switch, but with three control switches.

Outputs

Master Switch 1, 2, and 3 This output is connected to any off/on slider. It is only used when the node is the receiver. And the dependent input is connected [] Switch.

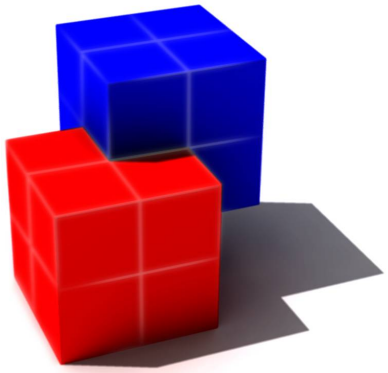
External Output

[] Switch Can be found in an External Dependency Input node.

Size = Position Snap

This dynamics node will snap an object to the global, grid equal to its independent x, y, z scales. Plus, a new step amount feature has been added in v1.5; allowing you to set the amount of steps per axis scale.

This can be a very useful node for building block type objects. By applying the node to an object's graph; then using the copy paste function, you can snap the objects end to end on any axis. This node currently does not work with rotated objects, as it will not snap properly.

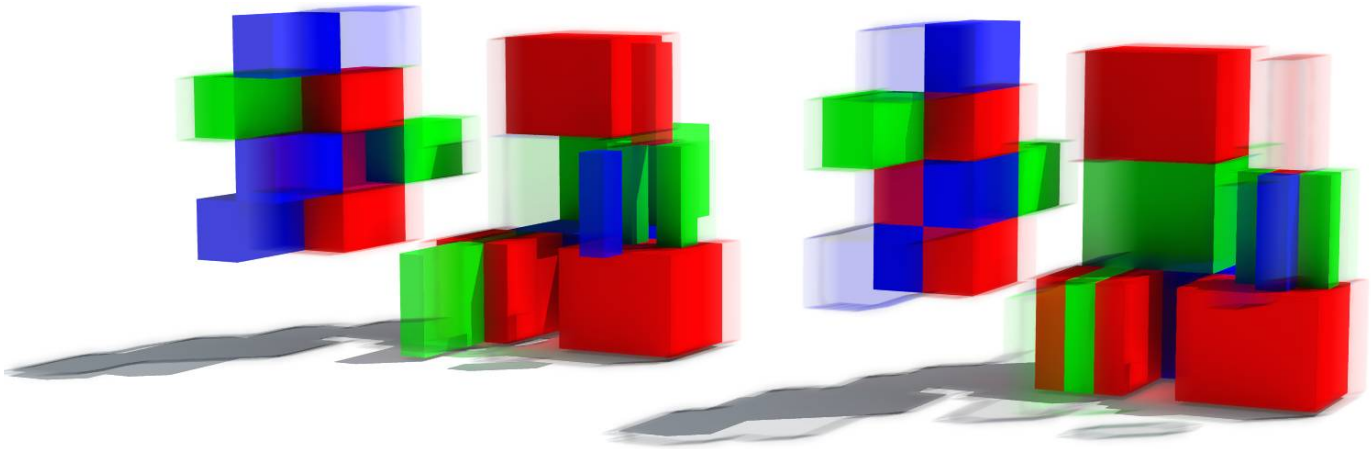


Example Steps: 2,2,2

Steps (added in node version 1.5)

x, y, z How many positional steps to equal the axis scale. 1,1,1 is = to total size.

Note: You can offset the start position of your object by using the Math>Vector Operations>Offset node. Connect the offset to the output link of the node, only use negative offset values.



How this node differs from the native Vue Grid Constraint:

The native node sets an equal value for the x, y, z axis. If you were to use the native node as a scale constraint, it would be limited to a single axis.

EXTREMELY IMPORTANT

Connecting and Removing this node from your object(s), MUST be made in the following order, or your object will be destroyed. Before connecting this node, save your scene. Below, the “Object Size Grid Constraint” Node, is referred to as the “Grid' node.

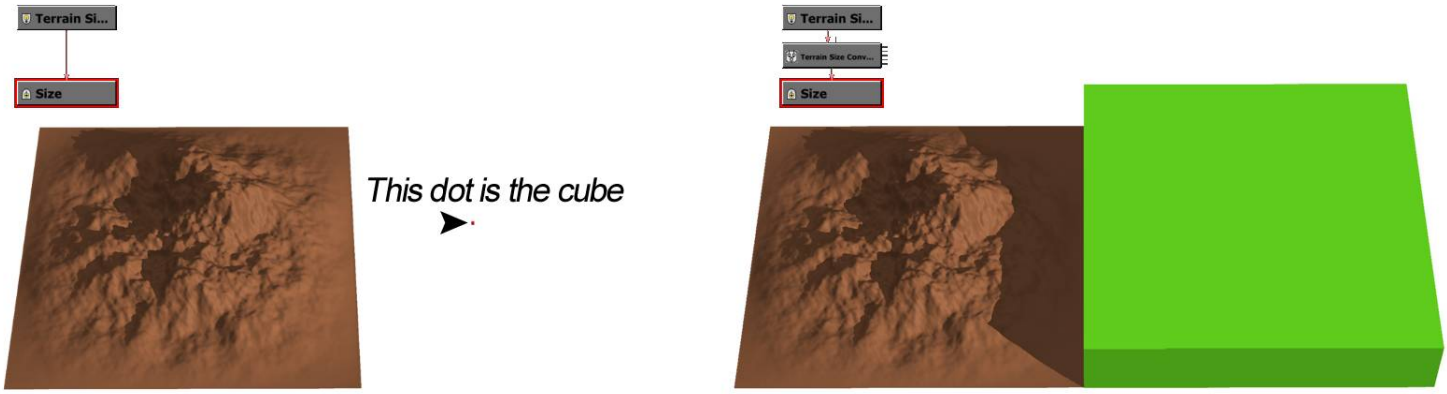
Connection Connect the Grid node's “Size” vector to the objects Size Input node. Then connect the object's Position output node, to the vector output (bottom) of the Grid node. Last connect the Grid node's “Position” vector to the object's Position Input node.

Node Removal Select the link from the object's Position output node to the Grid node, by clicking on it. Then press the delete key on the keyboard. Last, select the Grid node, by clicking on it, and press the delete key. This method should preserve the current position and keep the object intact.

Terrain Size Linker

Standard terrains have a different scale output than normal objects. Although the terrain will display the correct scale, linking this size output to other objects will prove otherwise. Manual vector entries in terrain graphs also need correcting.

Graph Example: Terrain Size -to- Object (Cube) Size



Left Image:
Without Terrain Size Conversion Linker

Right Image:
With Terrain Size Conversion Linker

A procedural terrain's x and y size are equal to normal object's vector. However, the z scale is almost completely random and constantly changing. The z size vector is dependent on the terrain's function, scale, image blend, and forced extension. The output is consistent with the Vue unit, but is then converted to a display or length unit. Although the Vue unit may remain the same, the conversion to length or display units changes with every terrain. Because of this, there is no way to link this scale with an object graph. Despite my many efforts to find a pattern to the this change, or find some way to control it; I have come to the conclusion, this is a current bug in vue or a limitation.

This conversion node may be used to control output vectors other than size, like terrain size to object position.

Inputs *(Vector 3)*

Size Connects to an external dependency (size)

Procedural Terrain Size = Self If controlling the size of a procedural terrain, connect this to the terrain's own size input. This must be connected before the metanode is connected to the output. This allows free scaling of the Z scale, since there is no way to accurately control the Z scale with a connected vector.

Outputs *(Vector 3) Match the object type with the node output.*

Object Converts a standard terrain's size vector for use with object vectors.

Terrain Converts an object's vector to control a standard terrains size.

Procedural Terrain Same conversion as the Object output, but the Z scale is null. Instead the Z scale of the "Procedural Terrain Size = Self" input is used; if using this output, make sure the self input in connected first.

Terrain from Procedural Terrain Converts a procedural terrain's size to a standard terrain size. The Z scale is then defined by the manually entered "Z Size" from the node, this is the only time the Z Size is used.

Terrain Pixel Dimensions

These number amounts can be found in the terrain editor. Example: 512x512

X Pixel Size

Y Pixel Size

Input – Procedural Terrain Size

Z Size This is the vector amount to define the Z scale, when a Procedural Terrain Size is the Size input, and used to control a standard terrain. (Terrain from Procedural Terrain, output)
Note: Because of the way Vue outputs terrain Z scale, for both procedural and standard, this amount may not be exact; however, it will be really close.

Terrain Size

0=Input 1=Terrain Vector At 0 the node uses the "Size" input, at 1 the node uses the Terrain Vector.

Terrain Vector Allows you to use a manual entry of a vector unit to control the output. If this vector is extracted, it can be used the same as the "Size" input. You could then use the above slider to switch between inputs.

Vector Lock

This node allows you to either constrain movement on an axis, or lock in vector for each axis.
All movement is free until an axis is locked, which can be done by using the Lock Axis sliders. Once the axis is locked, it will use the vector information from the Locked Position - Vector.

If you want to lock in the current vector, type in the vector or length unit into the Locked Position - Vector. (This information can be found in the numerics tab when the object selected.) Then move the Lock Axis slider to the locked position. The axis will then be locked to the entered vector of the given axis.

You can unlock a vector at any time by moving the slider to the open position. The set lock values will remain until the vector axis is modified.

About the unit system:

This node includes a built in Length unit to Vector unit conversion slider. To use length units, move the slider to the "length" position, and enter in the scenes internal unit. You can now type in a direct unit when the internal unit is set to a number other than 1. This conversion will only apply to the locked position vector.

Unit Examples: Length Unit (Meters)

Internal Unit [.1]
(10) meters = (100) Vector Units

Internal Unit [1]
(100) meters = (100) Vector Units

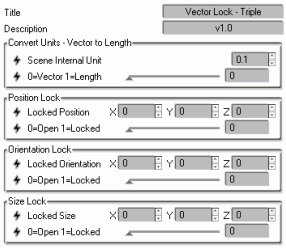
If you are extracting the Vector for the Locked Position, and connecting it directly to an object's vector, set the node to Vector Units.

If you are using this node to control Orientation (Rotation), set the node to Vector Units, conversion off.

Vector Lock - 3x3

Similar to to the "Vector Lock" node, but instead of locking individual axises, you can lock an entire vector with a slider, and lock entered values for Position, Orientation, and Size.

The Unit Conversion options of this node will not affect the Orientation inputs, since orientation doesn't change with the internal unit.



Single Axis Constraints

These axis constrains were designed primarily for graph connections, although they will work with any vector. They allow you to connect objects to other objects in the scene, and restrict a given vector axis. There are 3 Vector Constraint nodes, they are:

(x [Y] z)

[X] (y z)

(x y) [Z]

One node for each open axis. Although the inputs are full vectors and the unconnected axis becomes null.

The node itself is two nodes, one for each direction, (direction to input, or direction to output). These vectors are separated, and when using the Vector XYZ, input or output, the information is passed to the Vector non-XYZ inputs or outputs.

This is the information passed through each node type:

(x [Y] z) - x and z on a link, y null **(Vector = X,** null**, Z)**
y on a link, x and z null **(Vector =** null**, Y,** null **)**

(x y) [Z] - x and y on a link, z null **(Vector = X, Y,** null **)**
z on a link, x and y null **(Vector =** null**,** null**, Z)**

[X] (y z) - y and z on a link, x null **(Vector =** null**, Y, Z)**

NGOS - Numeric Graph Output Saver

NGOC - Numeric Graph Output Correction

In Vue 7+ Infinite, xStream, and possibly other versions; there is currently a bug (or limitation) manipulating the output of numbers from graphs. Until this is fixed these tools will correct this issue. The suggested tool for this is NGOS, because it is a work around that does not change the number value, just how it is transferred between functions and graphs. NGOC is a direct conversion; if this is a bug and is fixed, scenes or dependents using NGOC will be manipulated, however it can still be useful, and can always be removed.

At the moment this affects ALL external dependency input nodes, when receiving numeric information.

NGOS - Sender

NGOS - Receiver

Since vectors are not affected by this issue, NGOS will send number outputs as part of a vector, and then break them back down to numbers when received.

A couple of tools in this collection are using an integrated version of NGOS for their dependent controllers, which is why a vector is used to retrieve numeric information. This method of sending and receiving is quite nice, as it requires only one external dependency for every three numbers; so this ends up being a useful set of tools either way.

Sender
This node is used in the host graph, or where the numbers are coming from. NGOS - Sender has 9 number inputs, named 1 through 9. These inputs are combined into 3 vectors, and outputted with 3 internal custom dependencies. So all you need to do is connect the numbers you want to access in other graphs / functions.

The three dependency outputs are named:

[ ] Sender 1-3 ***[ ] Sender 4-6*** ***[ ] Sender 7-9***

There are also 3 vector outputs, in case you want to create your own custom dependencies. Named the same as above, but without [ ].

Receiver
After NGOS – Sender has been setup, you can load NGOS – Receiver into any other graph / function. If you have connected all 9 inputs, you will need 3 External Dependency Input nodes to connect the receiver, one for every 3.

There are three inputs at the top of the node, named:

1-3 ***4-6*** ***7-9***

Select the dependency outputs of the sender, in the external dependency input node.
The numbers are now accessible through the 9 outputs of the receiver, named 1 – 9.

NGOC – SR (Sender/Receiver)

Unlike NGOS, only one instance of this node is needed. It can be used as the sender or the receiver, but only use one instance of this node for the connection. Using it as a sender requires the least amount of work.

The internal custom dependency is called: [] NGOC – Sender. Only one number can be corrected at a time with this node.

NUMERIC OPERATIONS

Control Enhancement

Slider Sensitivity

This node allows you to adjust the sensitivity of any extractable slider number value. Allowing more control over slider tweak ranges.

If you are working with a slider with a range of 1 – 1000, and the output amount you are adjusting for is around the range of 100 – 200, then small movements in the slider range of 1 – 1000 will cause huge increases in output values, making it hard to tweak and adjust.

The Largest Feature range of a fractal is 1 – 1000, and this will be used for this example. Extract the Largest Feature from the fractal. Then load in the Slider Sensitivity MetaNode. Select the top of the link between the extracted node (constant node) and the fractal, and move it onto the Slider Sensitivity node to connect it. You can now delete the constant node that was created from the extraction.
Since we have a basic idea for the range we want to work with, select the Slider Sensitivity node, and set the “low - value” to 100 and the “high - value” to 200.

The Output slider in the node is your “tweak” slider, or the slider you will primarily use to set the output amount. The Sensitivity slider will reduce the range between the Low and High value you set. This means the low and high values do not need to be perfect. You can even set the same values as the original range, 1 – 1000 and turn down the sensitivity to decrease the range of the output slider.

Filters and Combiners

Fractal - Terrain Mixer

This tool will combine two fractals / noises (number inputs), creating an extremely useful terrain or material output. Several outputs are generated from the primary features of each number input. These outputs can be used to control color, ecosystem density, material distribution, or just about anything you can think of.

Inputs (Number)

Input 0 Connects to any number output.

Input 1 Connects to any number output, creating a mix with *Input 0*.

Reverse Inputs Slider Controls

Input 0 and 1 These sliders will invert the incoming number connection. The sliders range from 0 to 1, 0 being the normal input, and at 1 a reversed (opposite) input. At a slider value of .5, the input becomes nullified to a constant number of 0. Values between 0 - .5 and .5 - 1, will reduce to the total output, acting as a gain amount.

Outputs (Number)

Output 0 - 4 These five outputs are the mixed combinations of the 2 inputs. Any can be chosen to control a number output. Using a different output to control separate number outputs can produce some appealing results.

External Dependency Outputs (Number)

All 5 outputs from the mixer can be accessed throughout various functions in Vue. These outputs can be accessed by adding external dependency input nodes to any function, provided the mixer was created in a procedural terrain function, or an object graph. By select the host object as the dependency in the input node, the following outputs can be selected.

[] Output 0	[] Output 1	[] Output 2	[] Output 3	[] Output 4

Gradient Threshold

This filter is a little hard to explain, mainly because of the amount variations that can be outputted from it. Considering there are only 3 controls, it's really a matter of finding what you are looking for.

This filter works wonderfully for controlling ecosystem – size, presence, density, and color; it has also proved useful for controlling cloud density, material - distribution, highlights, color, and bump.
You can also create some amazing results by using several copies of this filter, linked in a chain to the number output.

Technically there are more than 3 controls, because a vast amount of variations can be controlled with the *gain* setting of a connected fractal.

Outputs

Output This is the primary number output of the fractal.

Reverse Output An inverted version of the main output.

Controls

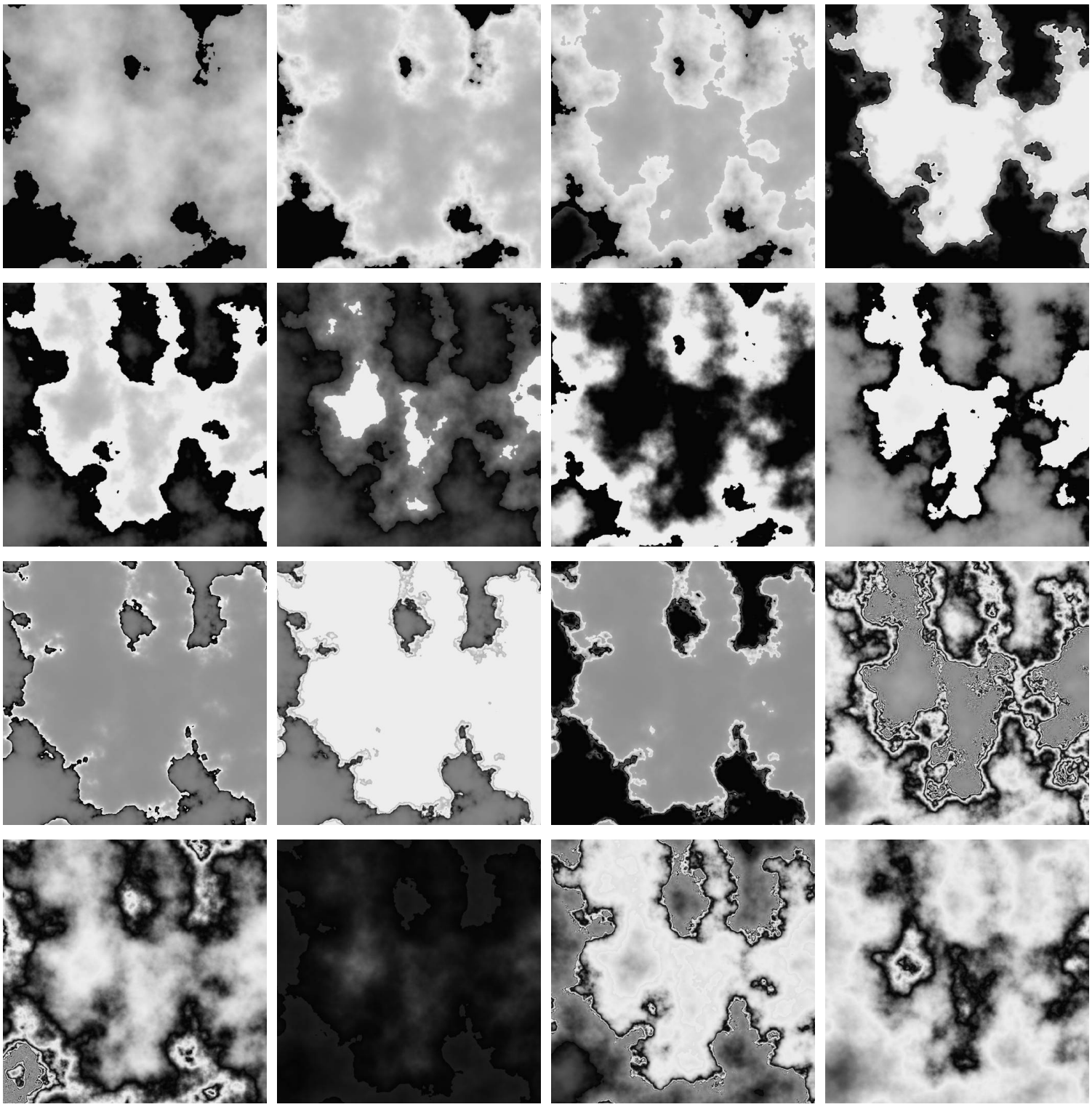
Falloff This control is almost completely random depending on the input. Sometimes it's a threshold, sometimes a gain, then a sine wave, and other times, like nothing you've ever seen. It's best not to try and understand what it does, just move it around and see what comes out of it. :)

Mode Changes an internal blending mode. This node was originally designed for just using the min mode, and this mode was not available to be changed in the controls. After a couple of modifications to the filter, the other modes became a valuable addition. These mode settings do not behave as they would when part of a combiner>blender node.

Offset Shifts the number output, moving through the noise.

Below is an assortment of outputs from this filter, all from the same input.

< This is the input that all of the following outputs where generated from. Just a simple perlin noise fractal. Only the gain setting in the fractal was modified between outputs.



Fractal / Terrain Filters

A selection of filters, useful for terrain altitude outputs and fractal filtering.

These filters and their affect on the output, rely heavily on the connected input amount (gain for fractal). All nodes have one control called, Gain Reversal.

Gain Reversal

Reduces the height or intensity of the connected number, before the filter. Higher amounts, both positive and negative, will reduce the input. Switching between positive and negative amounts will reverse, or invert the output.

Input Example

Mixed Elevation 1

Mixed Elevation 2

Plateau - Middle

Plateau - Ridged

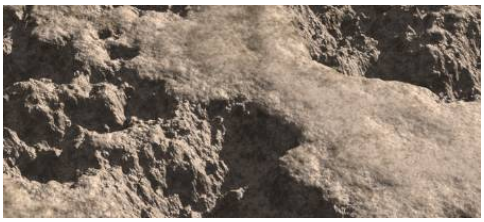
Plateau - Rough

Rivers 1

Rivers 2

Rivers 3

Sine Variation 1



Sine Variation 2

Sine Variation 3

Math

Cross Multiply = a

An extremely useful equation for comparing fractions.
If you are unfamiliar with cross multiplication, this is the structure of the formula, solving for a:

a(x) c
b d b x c / d = a(x)

The calculated number output is a.

Cross Multiply = b

Same as Cross Multiply but solving for b instead of a.

An extremely useful equation for comparing fractions.
If you are unfamiliar with cross multiplication, this is the structure of the formula, solving for b:

a c
b(x) d a x d / c = b(x)

The calculated number output is b.

Cross Multiply = b c

When only [a] and [d] are known. Solve for *y*.

a c(*y*)
b(y) d y = Square Root (a x d)

The calculated number output is for c and b, but only as one output since they are equal.

TEXTURE MAP OPERATIONS

Texture Map Setups

Image Channel Creator

When additional outputs are needed from a single texture map, for instance, using a texture map for color and bump, additional filters are often needed. This node eliminates the setup of this node group.
The main channel outputs from this node are:

Color	Bump	Highlight	Diffuse	Alpha	Alpha 2
[Color]	*[Number]*	*[Number]*	*[Number]*	*[Number]*	*[Number]*

Each channel has its own editable filter, with default values that generally work for most images.

Three additional controls are also available, and will affect all outputs: Brightness, Contrast, and Gamma. An additional output is available to bypass these corrections for the color channel; this output is named, Color – Correction Bypass.

Two primary Alpha / Transparency output are available. Alpha 2 creates a greater defined clip of the grayscale output. This output can be fine tuned with the ”Alpha 2 – Threshold” and “Alpha 2 – Falloff” sliders. The threshold increases or decreases the amount of transparency. The falloff adjusts the edge blending of the clip, from a harsh transition to a smooth one.

If you would like an inverted or opposite channel output, you can use the additional outputs. Every channel except the color channel has a “ (Channel) – Reversed” output. For instance, if you want to invert the current highlight areas from the highlight channel, you can switch the connection from the “Highlight” output to the “Highlight – Reversed” output.

Anytime an image is used to define the bump channel of a material, the default depth amount of 1 is normally too high (Bump Depth can be found in the Bump tab of the Advanced Material Editor). It is suggested you start with a value of .10 and tweak from there. Sometimes a value as small as .01 is needed to produce a clean looking bump. Using the Brightness, Contrast, and Gamma sliders in the node, to make overly dramatic changes, which could be easily made in an image editing program; may cause the channel outputs to over-sample, and max out past the desired levels. This is of course, true when ever using any of the similar filters in vue to affect a number output.

Matte Painter

Vue 7.5+ Recommended

Matte painter is an integrated camera projected texture map setup. It is intended for use with a one map setup. It includes the Aspect Ratio node built-in to control the image size.

This node includes all the features of the Texture Map node, with two new groups:

Camera Projection

Projector Choose the camera to project the image from.

Matte Aspect Ratio or Pixel Size

Width X - Height Y This is the ratio and size of the Matte – Image, and not necessarily the render output ratio.

Clip Outside Projection *(number)*

This output can be connected to the Alpha output node of the material function. It will cut out all parts of the image not contained within the camera projection.

Camera Master Controller

Allows the node to be connected to the Camera Master Controller node to control the projection. To use, add the external dependency input node, and select the camera with the master controller loaded, select camera projection as the dependent.

The following settings will be controlled by the master controller and will no longer work in the node.
Aspect Ratio – Width and Height
Image Tweaker – Scale
Camera Projection – Projector

Simple Image Mapper

If you are using a set of images to control a material, this node allows easy building of the function by creating the initial setup.

Channel outputs are:

Color	Bump	Highlight	Alpha
[Color]	*[Number]*	*[Number]*	*[Number]*

Each image used for the four channel output is linked by the texture's:

Scale	Origin	Image Offset
[Texture Coordinate]	*[Vector]*	*[Texture Coordinate]*

The defined setup is as follows, and the same for all texture maps:

Mip-Mapping	X Tiling	Y Tiling	Image Interpolation
Allowed	*Repeat*	*Repeat*	*Bi-linear*

Mapping modes can be controlled in the main node by selecting the edit button.

Conversions, Dynamics, and Controllers

Aspect Ratio

Used to control an extracted “Aspect Ratio” number amount from the “Camera Projection” texture coordinate input node.

Setting the Width [X] and Height [Y] amounts in the aspect ratio node, will fit the image in the connected Texture Map node proportionately onto the given object. Width and height can also be entered as a pixel amount.

With the aspect ratio set to the image pixel dimension or aspect ratio at render time, it will stay fixed regardless of the render output aspect ratio.

If rendering a material to project as an image on to an object, use the aspect ratio used to render the image to control the aspect ratio width and height.

You can also use the “*Matte Painter*” and “*Matte Painter – 2 Layer*” nodes for integrated aspect ratio and projection solutions.

Camera Projection

Vue 7.5+ Recommended

This is an improved version of the Camera Projection texture Coordinate node. The Camera Projection node is a texture coordinate node used to replace the UV Coordinate input node to a Texture Map's input.

This node features a built-in Aspect Ratio control. It can be used to control multiple camera projected map setups.

Aspect Ratio (see the “aspect ratio” node for details.)

Scale Used to scale the image outside the projection area. (1 , 1) is the default setting and keeps the image proportionate to the aspect ratio within the camera frame.

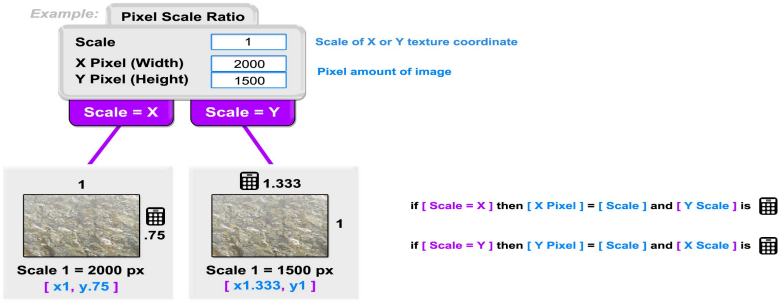
Image Offset This setting does not affect the projection output. It is used to control the extracted image offset of a Texture Map or multiple texture maps. When several textures are connected to this node, you can control all offsets with this setting, without having to change every texture node.

Projector Used to choose the camera to project the image from.

Pixel Ratio = Scale Ratio

Textures loaded into most software applications, including Vue, are initially scaled to 1x1 and made square.
This node will calculate the texture scale, to match the scale and proportions of an image based on the width and height pixel amount, x and y.

Two outputs are available, so you can chose to define the scale for X or Y. Both of these nodes will output the correct alternate value, and only one needs to be chosen.


Example:
Pixel Scale Ratio
Scale
1
Scale of X or Y texture coordinate
X Pixel (Width)
2000
Y Pixel (Height)
1500
Pixel amount of image
Scale = X
Scale = Y
1
.75
Scale 1 = 2000 px
[x1, y.75]
1.333
1
Scale 1 = 1500 px
[x1.333, y1]
if [Scale = X] then [X Pixel] = [Scale] and [Y Scale] is
if [Scale = Y] then [Y Pixel] = [Scale] and [X Scale] is

Object Plane Texture Controller

This node is for correcting scale differences, and convert object plane xy size vectors, to texture scale. It also links the xy position origin to any object plane selected within the node.

For this node to work correctly you must use the World Mapping mode on the material, and not make any changes to the origin or offset in the Advanced Material Editor tabs.

If you are using a material scale other than 1, entering in the material scale in the node will correct the texture's scale, making it equal to the object.

Controller Position and Scale, allows you to chose which object plane in the scene you would like to control the texture.

Position controls the vector output, and Scale controls the texture coordinate output.

Because controlling the rotation of a texture can not be controlled by an object, there is no way to control the rotation of the texture with this node.

VECTOR OPERATIONS

Composer Modifiers

Composer 2

This a remake of the default Vue Composer 2 texture coordinate node, which combines two number inputs into a single texture coordinate vector output. This version allows you to input a number directly into the node, rather than connecting it to a number output, although the connection to number outputs can still be made by extracting the x and y inputs.

Composer 2 - Add Number

Just like the Composer 2 node, you can enter number valued directly into this node, to define the x and y vector. The difference with the "- Add Number" node is, you can use the input connections (on the top of the node), to connect to any number output, then add to the value with the internal input boxes, or extractable links.

Composer 3

This a remake of the default Vue Composer 3 vector node, which combines three number inputs into a single vector output. This version allows you to input a number directly into the node, rather than connecting it to a number output, although the connection to number outputs can still be made by extracting the x y or z inputs.

Composer 3 - Add Number

Just like the Composer 3 node, you can enter number valued directly into this node, to define the x y and z vector. The difference with the "- Add Number" node is, you can use the input connections (on the top of the node), to connect to any number output, then add to the value with the internal input boxes, or extractable links.

Composer 3 x 3

This node is essentially 3 composer nodes in one. The number inputs and compiled vectors are named for the three primary object vector deformations, Position, Orientation, and Size. This node simplifies the setup for graph like functions when controlling multiple vectors with number inputs.
The axis deformation naming is primarily there as a guide, please do not feel restricted by them, because they can be used for any vector. If you find this distracting, please use the "Composer 3 x 3 simple" node.

Composer 3 x 3 - ni

Exactly the same as Composer 3x3, but the ni version uses the conventional Vue numeric input naming system, with the vector outputs named a, b, and c.

Conversion Operations

Internal Unit Vector Correction

Allows you to use numeric vector values equal to the main length unit, rather than being divided or multiplied by the internal unit. *If the internal unit is set to 1, then this node isn't needed.*
Node example: If you are entering in a manual value to control position or size within a function or graph, say a position origin of 10, and using the default internal unit, (0.1). 10 by default would equal 1 length unit. By using this node, you can type in 10, and the output will equal 10 length units.

Length to Vector

This is a replacement node for the “Internal Unit Vector Correction” node.

A conversion slider has been added to the node. Allowing you to turn off the conversion at any time. The slider is called “0=off 1=Convert.” When the slider is at 0 the node is turned off, and vector units are used with no conversion made to the inputs. When the slider is at 1, the node will convert entered length units to a vector output unit, by using the internal unit.

A number input and output has also been added to the node.

In addition to the new controls, two built in Dependencies are now available. These dependencies are a direct connection to the “Scene Internal Unit” and the “0=off 1=Convert” slider. By adding an external dependency input node to other functions and graphs in the scene, you can select these two outputs. The outputs can be used to control duplicates of the node in other objects graphs or functions. This can be used as a global switch for all objects using the conversion.

The dependency will be found with the name:
[ ] Conversion Dependent
.
You will need to use the Conversion Dependent node to output the numeric information correctly.

If using the master switch option, be sure to select the same object's dependents through out the scene. And never connect the master object to these dependents, doing so could create an endless loop of information and cause problems.

The extra dependencies will not be available for a master switch within materials. You may control material functions with the dependency, but the External Dependency will only display the output if the node is hosted in an object graph, or terrain procedural. Because of this, you may need to add an object into the scene to host the master node within the graph. Be sure to connect the host node to an output of the graph, if it is not connected, it may get removed. Since the object is just a control, the output node wont matter. Also consider using the “hide from render” and “ignore when populating ecosystems” options for control objects.

Vector to Brightness

This node coverts a vector into grayscale brightness number data.

Decomposer Modifiers

Decomposer 2 - Add Number

The traditional Decomposer 2 node splits apart a texture coordinate vector into 2 number outputs, for x and y, input 0 and input 1 respectively. This node has two internal number entry fields which add to the split vector. These inputs can also be extracted and connected to any number output.

The two outputs on the bottom of the node are the split vector as x and y number outputs, with the added values from the x and y number inputs.

Decomposer 3 - Add Number

The traditional Decomposer 3 node splits apart a vector into 3number outputs, for x, y, and z, input 0, input 1, and input 2 respectively. This node has three internal number entry fields which add to the split vector. These inputs can also be extracted and connected to any number output.

The three outputs on the bottom of the node are the split vector as x, y, and z number outputs, with the added values from the x, y, and z number inputs.

Decomposer 3 x 3

This node is essentially 3 decomposer nodes in one. The vector inputs and number outputs are named for the three primary object vector deformations, Position, Orientation, and Size. This node simplifies the setup for graph like functions when splitting multiple vectors into number outputs.
The axis deformation naming is primarily there as a guide, please do not feel restricted by them, because they can be used for any vector. If you find this distracting, please use the “Decomposer 3 x 3 simple” node.

Decomposer 3 x 3 - Simple

Exactly the same as Decomposer 3x3, but this node makes no assumptions about its usage. The outputs and inputs of this node are named 1, 2, and 3, instead of Position, Orientation, and Size. If you are used to the conventional Vue naming system, please use the “Decomposer 3 x 3 ni” node.

Decomposer 3 x 3 - ni

Exactly the same as Decomposer 3x3, but the ni version uses the conventional Vue numeric input naming system, with the vector inputs named input 0, input 1, and input 2. The number outputs are named as x, y, and z, with the input number displayed before the axis.

Offset Operations

Offset Split Vector Number

This node allows you to offset or control a vector by using 3 number links. These links can be extracted and connected to any number output to control a specific axis.

Offset Split Vector Number - add

This node preforms the same action as the “Offset Split Vector Number” node, but in the opposite direction.

re(de)composer Group

The re(de)composer MetaNodes are improved versions of the native Vue, composer and decomposer math nodes. Composer and Decomposer nodes break up a vector or texture into three number input and outputs (vector x,y,z), or two number inputs and outputs (texture x,y). In most cases these nodes are loaded separately, and almost always used together.

The improvements to this node, are primarily the ability to load in one node to handle all types of the composer and decomposer nodes.

The native Vue nodes are:
Composer 3, Combines 3 number inputs into a vector output.
Composer 2, Combines 2 number inputs into a texture coordinate vector output.
Decomposer 3, Splits a vector into 3 number outputs
Decomposer 2, Splits a texture coordinate vector into 2 number outputs.

As of the MetaNode Collection version 1.5, all composer integrations are extractable number inputs.

re(de)composer 2 - 3

This node is a combination of all 4 native nodes. It works as both a composer and decomposer, with the ability to convert both texture coordinates and vector values.

All channels work independently, so, the node can be used with multiple links with no relationship to each other.

re(de)composer 2

A combination of the Composer 2 and Decomposer 2 nodes, this node will split a connected texture coordinate vector into 2 number outputs, and / or combine 2 number inputs into a single texture coordinate vector. Each operation is independent.

re(de)composer 3

A combination of the Composer 3 and Decomposer 3 nodes, this node will split a connected vector into 3 number outputs, and / or combine 3 number inputs into a single vector. Each operation is independent.

re(de)composer 3 x 3

This MetaNodes is for splitting three vector inputs into 9 number outputs, and / or combining 9 number inputs into 3 vector outputs. The node was designed for controlling Position, Orientation, and Size, but can be used to control any three vectors in a function and/or graph.
Each set of vectors and its related 3 numbers work independently.

Vector Math

Divide Vector

Replaces the Previous nodes – Divide Vector by number and by vector. The number will divide the vector equally, the vector will divide per axis. If only using one option, the other(s) need to be set to a value of 1.

Divide Vector - by Number

Divides a vector by a single number, and applies the operations to all values.

Divide Vector - by Vector

Divides a vector by x,y,z vector values. Divides the x,y,z vector by the corresponding x,y,z vector. This vector can be entered as numbers in the main node, or extracted.

Vector Splitters

Split Vector from Input

This node will break up a vector from an input. The values remain as a vector, but is constrained to either the x, y, or z. For example:

If a vector of 40, 239, 34 is connected to this node, these values are converted to 3 vectors, and show as such, (x being a null value, or in some cases 0)
X (40, x, x)
Y (x, 239, x)
Z (x, x, 34)

This is a very useful node for creating total axis constraints.

Split Vector to Output

This node breaks down three vector inputs into 1 vector output. The output consists of input 0 (Named x) from the first vector, input 1 (Named y) from the second vector, and input 2 (Named z) from the third vector.

Split Vector In Out

This node combines the two nodes, “Split Vector from Input” and ““Split Vector to Output”, into one node, eliminating the need for the other two.

Split Vector In Out (Pass-through)

This node combines the two nodes, “Split Vector from Input” and ““Split Vector to Output”, into one node, and adds the values together.

NODE INDEX

Node:	Version	Operation	Short Description	Locked?
2 Map Color Brightness Variation	2.0	Color	Creates a fractal color distribution.	*Locked*
Advanced Sphere Zone	1.0	Dynamic Controller	Uses a sphere to control a numeric volume.	*Locked*
Aspect Ratio	1.0	Numeric Calculation	Calculates ratio from width and height.	*Locked*
Camera Master Controller	1.0	Dynamic Controller	Universal camera and texture projection controller.	*Locked*
Camera Orbiter	1.0	Dynamic Controller	Turns any camera into an orbital camera.	*Locked*
Camera Projection	1.0	Texture Controller	Integrated camera projection input and aspect ratio calculator.	*Locked*
Color Darken	1.0	Color	Darkens and desaturates color.	*Locked*
Color Map Reversal	1.0	Color	Reverses the direction of a color map, with additional adjustments.	*Locked*
Color Map Switcher	1.0	Color	Controls ratio of 2 color maps with an internal blender.	*Locked*
Composer 2	1.0	Texture	Turns two numbers into a texture coordinate, using extractable connections rather than inputs.	Editable
Composer 2 – Add Number	1.0	Texture	Adds two number inputs and two extractable inputs together to create a texture coordinate.	Editable
Composer 3	1.0	Vector	Turns three numbers into a vector, using extractable connections rather than inputs. Uses XYZ naming system.	Editable
Composer 3 – Add Number	1.0	Vector	Adds three number inputs and three extractable inputs together to create a vector.	Editable
Composer 3 x 3	1.5	Vector	Turns 9 number inputs into 3 vector outputs, using extractable numbers as inputs. Uses XYZ naming system for Position, Orientation, and Size.	Editable
Composer 3 x 3 Simple	1.0	Vector	Turns 9 number inputs into 3 vector outputs. Uses a generic naming system.	Editable
Composer 3 x 3 - ni	1.5	Vector	Turns 9 number inputs into 3 vector outputs, using extractable numbers as inputs. Uses the native numeric naming system.	Editable
Conversion Dependent	1.0	Graph / Function Dependent	Links any function with length unit correction to the master control (Length to Vector).	*Locked*
Cross Multiply = a	1.1	Numeric Calculation	Fraction comparison a/b = c/d. Solves for a.	Editable
Cross Multiply = b	1.0	Numeric Calculation	Fraction comparison a/b = c/d. Solves for b.	Editable
Cross Multiply = b c	1.0	Numeric Calculation	Fraction comparison a/b = c/d. Solves for b and c.	Editable
Decomposer 2 – Add Number	1.0	Vector	Adds two extractable number inputs to a texture coordinate, and converts the output to, two number outputs.	Editable
Decomposer 3 – Add Number	1.0	Vector	Adds three extractable number inputs to a vector, and converts the output to, three number outputs.	Editable
Decomposer 3 x 3	1.0	Vector	Converts 3 Vectors into 9 number outputs. Uses XYZ naming system for Position, Orientation, and Size.	Editable
Decomposer 3 x 3 Simple	1.0	Vector	Converts 3 Vectors into 9 number outputs. Uses a generic naming system.	Editable
Decomposer 3 x 3 - ni	1.0	Vector	Converts 3 Vectors into 9 number outputs. Uses the native numeric naming system.	Editable
Distance = Size	1.0	Dynamic Vector	Calculates the distance between two object pivots, and uses the output to control size.	*Locked*
Divide Vector	1.0	Vector	Divides a vector by an internal or connected number or vector.	*Locked*
Divide Vector – by Number	1.0	Vector	Divides a vector by an internal or connected number.	*Locked*
Divide Vector – by Vector	1.0	Vector	Divides a vector by an internal or connected vector.	*Locked*
Dynamic Range Controller	1.5	Dynamic Controller	Dynamically controls any number in a function or graph, using an object with range/distance influence controls.	*Locked*
Easy Color Changer	1.0	Color	An advanced color correction tool.	*Locked*

Focal Length	1.0	Numeric Conversion	Divides a number input by two, producing an accurate focal length amount in mm.	Editable
Fractal – Terrain Mixer	1.0	Combiner Filter	Outputs 5 blend variations from two inputed noises or fractals.	*Locked*
Function Preview Scalar	1.0	Vector	Changes the scales of node previews in the function editor.	*Locked*
Gradient Threshold	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
HLS Shift Intensity Ranged	1.1	Color	HLS correction with a built in slider sensitivity control.	*Locked*
HLS Shift Super Intensity Ranged	1.1	Color	HLS correction with 2 built in slider sensitivity control.	*Locked*
Image Channel Creator	1.1	Texture Map	Produces material channel outputs from a texture map.	*Locked*
Improved Distance Constraint	1.0	Dynamic Vector Operation	Allows you to turn on/off the distance constraint on the individual axises.	*Locked*
Internal Unit Vector Correction	1.0	Vector Calculation	Converts entered length unit amounts, into vector output amounts, based on the scene's internal unit.	*Locked*
Leaf Color Randomizer	1.0	Color	Adds color variation to leaf textures, designed to stay close to realistic color tones.	*Locked*
Length to Vector	1.0	Vector Calculation	Converts entered length unit amounts, into vector output amounts, based on the scene's internal unit. With the ability to use it as a master control for other objects.	*Locked*
Master Switch	1.0	Graph / Function Dependent	Controls any Off/On Slider. Can be used as Master or Dependent.	*Locked*
Master Switch - 3	1.0	Graph / Function Dependent	Controls any 3 Off/On Slider. Can be used as Master or Dependent.	*Locked*
Matte Painter	1.0	Texture Map Setup	Texture map with integrated camera projection.	*Locked*
Matte Painter - 2 Layer	1.0	Texture Map Setup	Foreground and Background Texture Map setup with layer alphas, and Master Camera Controller inputs.	*Locked*
Mixed Elevation 1	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
Mixed Elevation 2	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
NGOS – Sender	1.0	Bug Work Around	Sends numeric outputs as vector outputs to the NGOS - receiver in other graphs.	*Locked*
NGOS – Sender	1.0	Bug Work Around	Receives vector information from NGOS – Sender and converts the data back into numeric data.	*Locked*
NGOC – SR	1.0	Bug Correction	Performs the calculation needed to fix numeric data sent through graphs.	*Locked*
Object Plane – Texture Controller	1.1	Dynamic Texture	Controls a texture map's position and size with an object plane.	*Locked*
Offset – Split Vector Number	1.0	Vector	Uses three extractable numbers to offset a vector.	Editable
Offset Split Vector Number – Add	1.0	Vector	Uses three extractable numbers to add to a vector.	Editable
Pixel Ratio = Scale Ratio	1.0	Texture Coordinate Calculation	Uses width and height data to correct texture scale when not proportionate.	*Locked*
Plateau – Middle	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
Plateau – Ridged	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
Plateau – Rough	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
Range Controller - Dependent	1.0	Graph / Function Dependent	Connects graphs to the Dynamic Range Controller.	*Locked*
re(de)composer 2 - 3	1.5	Vector	Combined Composer 2 and 3, with Decomposer 2 and 3. And all in one node. Each node runs separately.	Editable
re(de)composer 2	1.5	Vector	Combined Composer 2 and Decomposer 2. Each node runs separately.	Editable
re(de)composer 3	1.5	Vector	Combined Composer 3 and Decomposer 3. Each node runs separately.	Editable
re(de)composer 3 x 3	1.5	Vector	Three re(de)composers in one.	Editable
Rivers 1	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
Rivers 2	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
Rivers 3	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*

Scene Color Controller	1.0	Dynamic Color Controller	Master control node of several definable colors.	Editable
Simple Image Mapper	1.1	Texture Map	Setup of 4 linked texture maps, useful for texture linked Color, Bump, Highlight, and Alpha image maps.	Editable
Sine Variation 1	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
Sine Variation 2	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
Sine Variation 2	1.0	Numeric Filter	A fractal or noise (number input) filter.	*Locked*
Size = Position Snap	1.5	Dynamic Controller	Snaps an object's position to a grid created by an object's size. Snap step amount per axis is definable.	*Locked*
Slider Sensitivity	1.1	Numeric Controller	Creates a custom range slider with sensitivity controls. For the tweaking and fine tuning of fractals and color generators.	*Locked*
Split Vector from Input	1.0	Vector	Splits a vector input into three vector outputs, creates 2 null vectors per output.	Editable
Split Vector to Output	1.0	Vector	Combines 3 vector inputs into one vector output. Using x,y,z ; one axis from each input to create the output.	Editable
Split Vector In Out	1.0	Vector	Combined “Split Vector from Input “ and “Split Vector to Output,” operating independently.	Editable
Split Vector In Out - (Pass-through)	1.0	Vector	Combined “Split Vector from Input “ and “Split Vector to Output,” adding to each set of vectors.	Editable
Terrain Previewer Camera Controller	1.2	Dynamic Controller	Keeps a terrain fixed in the top camera view. With improved focal adjustments to reduce perspective distortion.	*Locked*
Terrain Size Linker	1.0	Vector Correction	Correctly links terrains and object scales.	*Locked*
Texture Map Outside Clip	1.0	Texture Map	Adds the ability to cut out a texture map size in the alpha channel.	*Locked*
Turbulent Color Variation	1.5	Color	Color variance generator with built in turbulence controls.	*Locked*
Vector Lock	1.0	Dynamic Vector	Allows the locking or opening of vector axises. Useful to avoid unwanted transformations.	*Locked*
Vector Lock 3x3	1.0	Dynamic Vector	Allows the locking or opening of 3 vectors. Designed for Position, Orientation, and Size. Useful to avoid unwanted transformations.	*Locked*
Vector to Brightness	1.0	Conversion	Converts a vector input to a number (brightness) output.	*Locked*
(x [Y] z)	1.0	Dynamic Vector	Axis constraint. Two vector inputs to control one output.	Editable
(x y) [Z]	1.0	Dynamic Vector	Axis constraint. Two vector inputs to control one output.	Editable
[X] (y z)	1.0	Dynamic Vector	Axis constraint. Two vector inputs to control one output.	Editable
Zone Dependent	1.0	Graph / Function Dependent	Controls the dependency output of Advanced Sphere Zone.	*Locked*

Collection Release Notes

v1.1 March 30th 2009

Improved refresh of both the openGL and output at render time.

Notes:
Changes made to Vue have caused improper renders and refresh of dependent graphs and functions; all MetaNodes with built in external dependencies, have had those dependencies removed.

Dependencies must now be connected to the MetaNode, rather than the connection being made within the node. This guide currently does not reflect these changes, and will be updated in the near future.

MetaNodes updated to v1.1:

Object Plane Texture Controller
Terrain Previewer – Camera Controller

The installation directories have been updated include the new node versions.
Also, an "updates" folder has also been added for quick access to the nodes and past versions.

v1.5 June 09th 2009

New Host Collection node available. asileFX_MetaNode_Collection.mnd
All nodes and tools can be accessed through this host master node. Use copy and paste to load the nodes into your functions / graphs.

Node / Tool Updates	Version	**Update Notes:**
Terrain Previewer – Camera Controller	1.2	Improved Organization.
2 Map Color Brightness Variation	2.0	Added a number input to control the 2 color maps. All controls are now extractable. Added new color "switcher" options.
Turbulent Color Variation	1.5	Improved Organization. All controls are now extractable.
HLS Shift Intensity Ranged	1.1	Improved Organization. Intensity now called "range" in the Intensity group.
HLS Shift Super Intensity Ranged	1.1	Improved Organization. Intensity now called "range" in the Intensity group. Super Sample now called "MetaRange."
Object Size Grid Constraint **Renamed: *Size = Position Snap***	1.5	Added steps control; the ability to control the snap amount per axis.
Slider Sensitivity	1.1	Improved Organization.
Cross Multiply **Renamed: *Cross Multiply = a***	1.1	Improved Organization.
Image Channel Creator	1.1	Improved Organization.
re(de)composers	1.5	Improved Organization all nodes in the group. Decomposers are connected to the top inputs, and all composer connections are made through extractable inputs. A much easier way of using this setup.
Composer 3 x 3 and ni	1.5	Improved Organization. All top number inputs replaced with extractable inputs.
Simple Image Mapper	1.1	Improved Organization. Origin, Image Offset, and Image Scale are now extractable.

v2.0 June 09th 2009

MetaNode Collection has been given a name, snowFlaKe

New Tools and Nodes	
Advanced Sphere Zone	Matte Painter - 2 Layer
Aspect Ratio	Mixed Elevation 1
Camera Master Controller	Mixed Elevation 2
Camera Orbiter	NGOS – Sender
Camera Projection	NGOS – Receiver
Color Darken	NGOC – SR
Color Map Reversal	Plateau – Middle
Color Map Switcher	Plateau – Ridged
Conversion Dependent	Plateau – Rough
Cross Multiply = b	Range Controller - Dependent
Cross Multiply = b c	Rivers 1
Distance = Size	Rivers 2
Dynamic Range Controller	Rivers 3
Focal Length	Scene Color Controller
Fractal – Terrain Mixer	Sine Variation 1
Function Preview Scalar	Sine Variation 2
Gradient Threshold	Sine Variation 3
Improved Distance Constraint	Terrain Size Linker
Leaf Color Randomizer	Texture Map Outside Clip
Length to Vector	Vector Lock
Master Switch	Vector Lock 3x3
Master Switch - 3	Zone Dependent
Matte Painter	+ Helpers and Setups